Modicon M340 con Unity Pro

Collegamento seriale
Manuale dell’utente

10/2013

35013363.07

www.schneider-electric.com

Schneider Electric

Questa documentazione contiene la descrizione generale e/o le caratteristiche tecniche dei prodotti qui contenuti. Questa documentazione non è destinata e non deve essere utilizzata per determinare l'adeguatezza o l'affidabilità di questi prodotti relativamente alle specifiche applicazioni dell'utente. Ogni utente o specialista di integrazione deve condurre le proprie analisi complete e appropriate del rischio, effettuare la valutazione e il test dei prodotti in relazioni all'uso o all'applicazione specifica. Né Schneider Electric né qualunque associata o filiale deve essere tenuta responsabile o perseguibile per il cattivo uso delle informazioni ivi contenute. Gli utenti possono inviarci commenti e suggerimenti per migliorare o correggere questa pubblicazione.

È vietata la riproduzione totale o parziale del presente documento in qualunque forma o con qualunque mezzo, elettronico o meccanico, inclusa la fotocopiatura, senza esplicito consenso scritto di Schneider Electric.

Durante l'installazione e l'uso di questo prodotto è necessario rispettare tutte le normative locali, nazionali o internazionali in materia di sicurezza. Per motivi di sicurezza e per assicurare la conformità ai dati di sistema documentati, la riparazione dei componenti deve essere effettuata solo dal costruttore.

Quando i dispositivi sono utilizzati per applicazioni con requisiti tecnici di sicurezza, occorre seguire le istruzioni più rilevanti.

Un utilizzo non corretto del software Schneider Electric (o di altro software approvato) con prodotti hardware Schneider Electric può costituire un rischio per l'incolumità personale o provocare danni alle apparecchiature.

La mancata osservanza di queste indicazioni può costituire un rischio per l'incolumità personale o provocare danni alle apparecchiature.

© 2013 Schneider Electric. Tutti i diritti riservati.

Indice

	Informazioni di sicurezza	**9**
	Informazioni su...	**11**
Parte I	**Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere**	**13**
Capitolo 1	**Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere**	**15**
	Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere	15
Parte II	**Installazione hardware per il Modbus Seriale e le comunicazioni in modalità carattere**	**17**
Capitolo 2	**Introduzione alle comunicazioni seriali**	**19**
2.1	Collegamento seriale sui processori BMX P34 1000/2000/2010/20102/2020	20
	Presentazione del collegamento seriale sui processori BMX P34 1000/2000/2010/20102/2020	20
2.2	Module BMX NOM 0200 a 2 porte RS-485/232	25
	Presentazione del modulo BMX NOM 0200 con 2 porte RS-485/232	26
	Apparecchiatura Modicon M340H (Hardened)	32
	Messa a terra dei moduli installati	33
	Installazione del modulo BMX NOM 0200	35
	Considerazioni sul cablaggio del modulo BMX NOM 0200	37
Capitolo 3	**Architetture di comunicazione seriale**	**39**
3.1	Architettura delle comunicazioni seriali per i processori BMX P34 1000/2000/2010/20102/2020	40
	Terminazione della linea Modbus e polarizzazione (RS485)	41
	Collegamento dei dispositivi Modbus (RS485)	43
	Connessione di un’apparecchiatura per terminale dati (DTE) (RS232)	46
	Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)	48
3.2	Architetture di comunicazione seriale per BMX NOM 0200	50
	Terminazione della linea Modbus e polarizzazione (RS485)	51
	Collegamento dei dispositivi Modbus (RS485)	53
	Connessione di un’apparecchiatura per terminale dati (DTE) (RS232)	55
	Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)	57

3.3 Cablaggio 60
Cablaggio 60
Parte III Implementazione software del Modbus Seriale e delle comunicazioni in modalità carattere 65
Capitolo 4 Metodologia di installazione 67
Introduzione alla fase di installazione 67
Capitolo 5 Comunicazione Modbus Seriale per i processori BMX P34 1000/2000/2010/20102/2020 69
5.1 Generalità 70
Informazioni sul Modbus Seriale 71
Prestazioni 72
Accesso ai parametri del collegamento seriale 74
5.2 Configurazione della comunicazione Modbus Seriale 77
Schermata di configurazione della comunicazione Modbus Seriale . . 78
Funzioni accesibili Modbus 80
Valori predefiniti dei parametri di comunicazione Modbus Seriale 81
Parametri Modbus legati all'applicazione 82
Parametri Modbus legati alla trasmissione 84
Parametri del segnale e della linea fisica in Modbus 86
5.3 Programmazione della comunicazione Modbus Seriale 88
Servizi supportati da un processore master per collegamento Modbus 89
Servizi supportati da un processore slave del collegamento Modbus . 96
5.4 Debug della comunicazione Modbus Seriale 98
Schermata di debug della comunicazione seriale Modbus 98
Capitolo 6 Comunicazione modalità carattere per i processori BMX P34 1000/2000/2010/20102/2020 101
6.1 Generalità 102
Informazioni sulla comunicazione in modalità carattere 103
Prestazioni 104
6.2 Configurazione di comunicazione in modalità caratteri 106
Schermata di configurazione della comunicazione in modalità carattere 107
Funzioni accessibili in modalità carattere 109
Valori predefiniti dei parametri di comunicazione in modalità carattere 110
Parametri di rilevamento fine messaggio in modalità caratteri 111
Parametri di trasmissione in modalità carattere 113
Parametri del segnale e della linea fisica in modalità carattere 115

6.3 Programmazione comunicazione in modalità caratteri 117
Funzioni di comunicazione in modalità carattere 117
6.4 Debugging della comunicazione in modalità caratteri 123
Schermata di debug della comunicazione in modalità carattere 123
Capitolo 7 Comunicazione Modbus Seriale per BMX NOM 0200 . 125
7.1 Generalità 126
Informazioni sul Modbus Seriale 127
Prestazioni 128
Accesso ai parametri del collegamento seriale 130
7.2 Configurazione della comunicazione Modbus Seriale 133
Schermata di configurazione della comunicazione seriale Modbus in un rack locale Modicon M340 134
Schermata di configurazione per la comunicazione seriale Modbus del modulo BMX NOM 0200 in derivazione X80 137
Funzioni accesibili Modbus 140
Valori predefiniti dei parametri di comunicazione Modbus Seriale . . . 141
Parametri Modbus legati all'applicazione 142
Parametri Modbus legati alla trasmissione 144
Parametri del segnale e della linea fisica in Modbus 146
Come impostare l'indirizzo slave Modbus del BMX NOM0200 senza Unity Pro? 148
7.3 Programmazione della comunicazione Modbus Seriale 150
Servizi supportati da un modulo master del collegamento Modbus . . 151
Servizi supportati da un modulo slave del collegamento Modbus 158
Informazioni dettagliate sulla modalità Expert Modbus 160
7.4 Debug della comunicazione Modbus Seriale 166
Schermata di debug della comunicazione Modbus 166
Capitolo 8 Comunicazione Modalità carattere per BMX NOM 0200 169
8.1 Generalità 170
Informazioni sulla comunicazione in modalità carattere 170
8.2 Configurazione comunicazione in modalità carattere 171
Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in un rack locale 172
Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in derivazione X80 175
Funzioni accessibili in modalità carattere 178

Valori predefiniti dei parametri di comunicazione in modalità carattere 179
Parametri di rilevamento fine messaggio in modalità carattere 180
Parametri di trasmissione in modalità carattere 182
Parametri del segnale e della linea fisica in modalità carattere 184
8.3 Programmazione comunicazione in modalità carattere 187
Funzioni di comunicazione in Modalità caratteri 188
Dettagli della modalità Expert in modalità Carattere 197
8.4 Debug della comunicazione in modalità carattere 200
Schermata di debug della comunicazione in modalità carattere 200
Capitolo 9 Diagnostica Modulo BMX NOM 0200 203
9.1 Diagnostica Modulo BMX NOM 0200 204
Diagnostica di un modulo BMX NOM 0200 205
Diagnostica dettagliata del canale di comunicazione. 207
Capitolo 10 Oggetti linguaggio di comunicazioni Modbus e modalità caratteri 209
10.1 Oggetti linguaggio e IODDT delle comunicazioni Modbus e in modalità carattere 210
Introduzione agli oggetti linguaggio delle comunicazioni Modbus e in modalità carattere 211
Oggetti linguaggio a scambio implicito associati alla funzione specifica dell'applicazione 212
Oggetti di linguaggio a scambio esplicito associati alla funzione specifica all'applicazione 213
Gestione degli scambi e dei rapporti con oggetti espliciti 215
10.2 Oggetti linguaggio e IODDT generici per i protocolli di comunicazione 218
Dettagli degli oggetti di scambi impliciti IODDT di tipo T_COM_STS_GEN 219
Dettagli degli oggetti di scambi espliciti IODDT di tipo T_COM_STS_GEN 220
10.3 Oggetti linguaggio e IODDT associati alla comunicazione Modbus. . . 222
Dettagli concernenti oggetti di linguaggio a scambi espliciti per una funzione Modbus 223
Dettagli sugli oggetti di scambio implicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT 224
Dettagli sugli oggetti di scambio esplicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT 225
Dettagli del linguaggio oggetti associato alla configurazione in modalità Modbus 228

10.4 Oggetti linguaggio e IODDT associati alla comunicazione in modalità caratteri 230
Dettagli concernenti oggetti di linguaggio a scambi espliciti per comunicazione in modalità caratteri 231
Dettagli relativi agli oggetti di scambio implicito dello IODDT di tipo T_COM_CHAR_BMX 232
Dettagli degli oggetti di scambio esplicito IODDT di Tipo T_COM_CHAR_BMX 233
Dettagli del linguaggio oggetti associato alla configurazione in modalità carattere 236
10.5 IODDT tipo T_GEN_MOD applicabili a tutti i moduli 238
Dettagli degli oggetti linguaggio dell'IODDT di tipo T_GEN_MOD . . . 238
10.6 Oggetti linguaggio e DDT dei dispositivi associati alla comunicazione Modbus 240
Nomi del DDT del dispositivo di comunicazione 240
Capitolo 11 Cambio dinamico di protocollo 243
Modifica del protocollo con i processori BMX P34 1000/2000/2010/20102/2020 244
Modifica del protocollo con il modulo BMX NOM 0200 246
Parte IV Avvio rapido: Esempio dell'implementazione del collegamento seriale 249
Capitolo 12 Descrizione dell'applicazione 251
Panoramica dell'applicazione 251
Capitolo 13 Installazione dell'applicazione con Unity Pro 253
13.1 Presentazione della soluzione utilizzata 254
Passi della procedura che utilizza Unity Pro 254
13.2 Sviluppo dell'applicazione 255
Creazione di un progetto 256
Dichiarazione delle variabili 261
Uso del modem 265
Procedura per la programmazione 267
Struttura della programmazione 268
Programmazione 271
Capitolo 14 Avvio dell'applicazione 281
Esecuzione dell'applicazione in modalità standard 281
Glossario 285
Indice analitico 293

Informazioni di sicurezza

Informazioni importanti

AVVISO

Leggere attentamente queste istruzioni e osservare l'apparecchiatura per familiarizzare con i suoi componenti prima di procedere ad attività di installazione, uso o manutenzione. I seguenti messaggi speciali possono comparire in diverse parti della documentazione oppure sull'apparecchiatura per segnalare rischi o per richiamare l'attenzione su informazioni che chiariscono o semplificano una procedura.

L'aggiunta di questo simbolo a un'etichetta di Pericolo relativa alla sicurezza indica che esiste un rischio da shock elettrico che può causare lesioni personali se non vengono rispettate le istruzioni.

Questo simbolo indica un possibile pericolo. È utilizzato per segnalare all'utente potenziali rischi di lesioni personali. Rispettare i messaggi di sicurezza evidenziati da questo simbolo per evitare da lesioni o rischi all'incolumità personale.

⚠ PERICOLO

PERICOLO indica una condizione immediata di pericolo, la quale, se non evitata, **può causare** seri rischi all'incolumità personale o gravi lesioni.

⚠ ATTENZIONE

ATTENZIONE indica una situazione di potenziale rischio che, se non evitata, **può provocare** morte o gravi infortuni.

⚠ ATTENZIONE

ATTENZIONE indica una situazione di potenziale rischio, che, se non evitata, **può provocare** infortuni di lieve entità.

AVVISO

Un ***AVVISO*** è utilizzato per affrontare delle prassi non connesse all'incolumità personale.

NOTA

Manutenzione, riparazione, installazione e uso delle apparecchiature elettriche si devono affidare solo a personale qualificato. Schneider Electric non si assume alcuna responsabilità per qualsiasi conseguenza derivante dall'uso di questo materiale.

Il personale qualificato è in possesso di capacità e conoscenze specifiche sulla costruzione, il funzionamento e l'installazione di apparecchiature elettriche ed è addestrato sui criteri di sicurezza da rispettare per poter riconoscere ed evitare le condizioni a rischio.

Informazioni su...

In breve

Scopo del documento

Questo manuale descrive il principio di implementazione hardware e software della comunicazione in modalità Caratteri e Modbus per i processori BMX P34 1000/2000/2010/20102/2020. Questo manuale descrive altresì l'installazione hardware e software del modulo di comunicazione BMX NOM 0200 per i PLC Modicon M340 e le derivazioni X80.

Nota di validità

Questa documentazione è valida dalla versione Unity Pro v8.0.

Informazioni relative al prodotto

⚠ AVVERTENZA

FUNZIONAMENTO ANOMALO DELL'APPARECCHIATURA

L'applicazione di questo prodotto richiede esperienza di progettazione e programmazione dei sistemi di controllo. Solo il personale in possesso di tali competenze è autorizzato a programmare, installare, modificare e utilizzare questo prodotto.

Rispettare la regolamentazione e tutte le norme locali e nazionali sulla sicurezza.

Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.

Parte I

Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere

Capitolo 1
Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere

Introduzione al Modbus Seriale e alle comunicazioni in modalità carattere

Generale

I collegamenti seriali dei processori BMX P34 1000/2000/2010/20102/2020 e del modulo BMX NOM 0200 supportano due protocolli di comunicazione:

- Modbus seriale
- Modalità carattere

Protocollo Modbus

Modbus è un protocollo standard con le seguenti proprietà:

- stabilisce la comunicazione client/server tra moduli diversi all'interno di un bus o collegamento seriale. Il client viene identificato dal master e i moduli slave rappresentano i server
- è basato su una modalità di scambio dati composta da richieste e risposte e offre servizi tramite diversi codici funzione.
- stabilisce un mezzo per lo scambio di frame da applicazioni di tipo Modbus in due tipi di codice:
  - Modalità RTU
  - Modalità ASCII

La procedura di gestione di scambio si presenta come segue:

- solo un dispositivo può inviare dati sul bus;
- gli scambi vengono gestiti dal master. Solo il master può iniziare gli scambi. Gli slave possono inviare messaggi solo a seguito di richiesta;
- nel caso di uno scambio errato, il master ripete la richiesta. Lo slave al quale viene inviata la richiesta è dichiarato assente dal master se non risponde entro un determinato periodo di tempo.
- lo slave, se non comprende o non è in grado di elaborare la richiesta, invia una risposta di eccezione al master. In questo caso, il master può ripetere o meno la richiesta.

Tra master e slave sono possibili due tipi di dialogo:

- Il master invia una richiesta a un numero di slave specifico e attende la risposta.
- Il master invia una richiesta a tutti gli slave senza attendere risposta (il principio di trasmissione generale).

Comunicazione in modalità carattere

La modalità carattere è una modalità punto a punto di scambio dati tra due entità. A differenza del Protocollo Modbus, non stabilisce comunicazioni di collegamento seriale strutturate gerarchicamente né offre servizi tramite codici di funzione.

La modalità carattere è asincrona. Ogni elemento di informazione testuale viene inviato o ricevuto carattere per carattere a intervalli di tempo irregolari. Il tempo impiegato dagli scambi può essere determinato dalle proprietà seguenti:

- uno o due più caratteri di fine frame
- timeout
- numero di caratteri.

Parte II

Installazione hardware per il Modbus Seriale e le comunicazioni in modalità carattere

In questa sezione

Questa sezione fornisce un'introduzione all'installazione hardware per comunicazioni seriali Modbus e in Modalità carattere.

Contenuto di questa parte

Questa parte contiene i seguenti capitoli:

Capitolo	Titolo del capitolo	Pagina
2	Introduzione alle comunicazioni seriali	19
3	Architetture di comunicazione seriale	39

Capitolo 2
Introduzione alle comunicazioni seriali

Argomento del capitolo

Questo capitolo spiega come si svolgono le comunicazioni seriali sui processori BMX P34 1000/2000/2010/20102/2020 e sul modulo BMX NOM 0200.

La tabella di seguito fornisce una rapida panoramica delle due possibilità di implementazione delle comunicazioni con collegamento seriale:

Con la porta integrata della CPU	Con il modulo di comunicazione BMX NOM 0200
- Velocità di trasmissione limitata - Linee seriali non isolate - Fornitura di alimentazione elettrica alle apparecchiature terminali	- Maggior numero di canali di comunicazione disponibili - Gestione di segnali modem RS232 specifici - Velocità di trasmissione più elevata - Due linee seriali RS485 isolate

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
2.1	Collegamento seriale sui processori BMX P34 1000/2000/2010/20102/2020	20
2.2	Module BMX NOM 0200 a 2 porte RS-485/232	25

Sezione 2.1
Collegamento seriale sui processori BMX P34 1000/2000/2010/20102/2020

Presentazione del collegamento seriale sui processori BMX P34 1000/2000/2010/20102/2020.

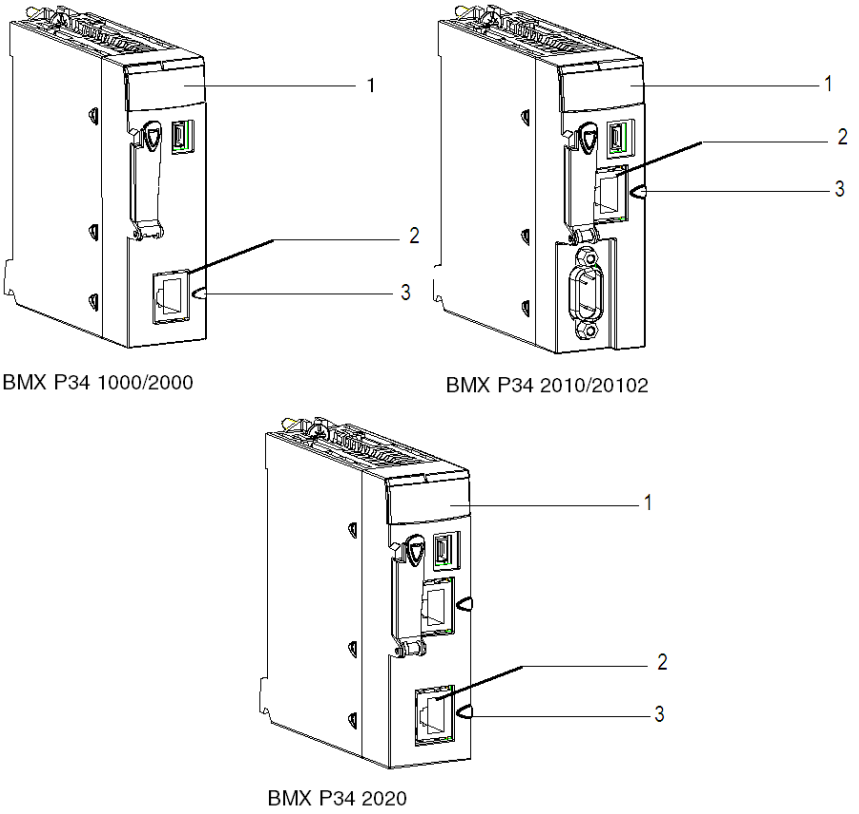
Informazioni generali

I seguenti processori hanno un canale di comunicazione integrato dedicato alle comunicazioni seriali e permettono la comunicazione tramite il collegamento seriale:

- BMX P34 1000/2000/2020,
- BMX P34 2010/20102.

Introduzione ai processori

L'illustrazione seguente mostra le caratteristiche fisiche dei processori BMX P34 1000/2000/2010/20102/2020:



BMX P34 1000/2000

BMX P34 2010/20102

BMX P34 2020

Questi processori sono composti dai seguenti elementi:

Riferimento	Descrizione
1	LED di stato del processore nella parte anteriore
2	Canale integrato dedicato (canale 0) al collegamento seriale.
3	Anello di identificazione della porta seriale (nero)

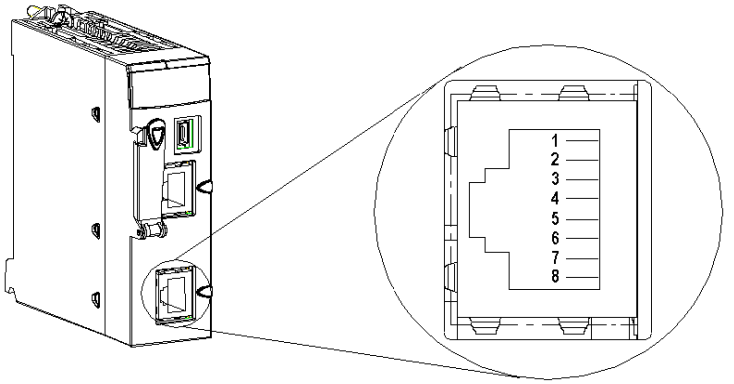
Diagnostica visuale della comunicazione seriale

Lo stato della comunicazione seriale è indicato da un LED SER COM giallo situato sul lato frontale del processore:

- LED lampeggiante: La comunicazione seriale è in corso.
- LED spento: la comunicazione seriale non è in corso.

Introduzione alla porta seriale

Nella seguente figura è illustrata la porta seriale RJ45:



Il connettore RJ45 ha 8 pin. I pin utilizzati variano in base al collegamento fisico impiegato.

I pin utilizzati dal collegamento seriale RS232 sono:

- Pin 1: segnale RXD
- Pin 2: segnale TXD
- Pin 3: segnale RTS
- Pin 6: segnale CTS
- Pin 8: messa a terra collegamento seriale potenziale (0 V)

I pin utilizzati dal collegamento seriale RS485 sono:

- Pin 4: segnale D1
- Pin 5: segnale D0

Il pin 7 viene utilizzato esclusivamente per fornire alimentazione a interfacce uomo macchina o piccoli dispositivi tramite il cavo di collegamento seriale:

- Pin 7: alimentazione collegamento seriale: 5 VDC/190 mA

Caratteristiche dettagliate

Caratteristiche DC:
- Consumo stabilizzato massimo: 190 mA,
- Tensione minima sul connettore CPU per 190 mA: 4,9 V,
- Tensione massima sul connettore CPU per 190 mA: 5.25 V,
- Tensione massima sul connettore CPU con carico nullo: 5.5 V.

Caratteristiche AC:
- Carica condensatore: (su 5 V)
  - Condensatore in ceramica da massimo 1 µF
  - tantalio 10 µF
- Avvio carica della pompa: (su 5 V)
  - Condensatore in ceramica da 4 x 1 µF
  - tantalio 2 x 10 µF

NOTA: i collegamenti RS232 a quattro fili, RS485 a due fili e RS485 a due fili con alimentazione utilizzano tutti lo stesso connettore femmina RJ45. Solo il cablaggio di segnale è diverso.

Specifiche della linea elettrica

Le linee RS232 e RS485 non sono isolate.

In caso di messa a terra non equipotenziale tra le apparecchiature collegate (cavi di almeno 30 m), è necessario usare un modulo isolatore TWDXCAISO in modalità RS485.

La polarizzazione della linea RS485 è integrata nel PLC e automaticamente attivata o disattivata dal sistema in base alla configurazione scelta nella schermata Unity Pro :
- Master Modbus: La polarizzazione della linea è attivata.
- Slave Modbus: la polarizzazione della linea è disattivata.
- Modalità caratteri: la polarizzazione della linea è disattivata.

La polarizzazione non è compromessa dal cambio dinamico di protocollo. Il valore dei resistori di polarizzazione è 560 ohm.

In modalità RS232 non è richiesta alcuna polarizzazione.

Non è presente alcuna linea di terminazione integrata.

Specifiche di canale

Il canale di questi processori comprende:
- Un'interfaccia fisica RS485 non isolata.
- Un'interfaccia fisica RS232 non isolata.
- Tipi di comunicazione Modbus Seriale (ASCII e RTU) e Modalità caratteri.

Le specifiche di collegamento per i due protocolli sono:

	Modbus seriale / RS485	**Modbus Seriale / RS232**	**Modalità caratteri / RS485**	**Modalità caratteri / RS232**
Tipo	Master/Slave	Master/Slave	Half Duplex	Full Duplex
Flow	19200 baud. I parametri possono essere impostati da 300 baud a 38400 baud.	19200 baud. I parametri possono essere impostati da 300 baud a 38400 baud.	9600 baud. I parametri possono essere impostati da 300 baud a 38400 baud.	9600 baud. I parametri possono essere impostati da 300 baud a 38400 baud.
Numero di apparecchiature	32	32	_	_
Indirizzi slave autorizzati	da 1 a 247	da 1 a 247	_	_
Lunghezza massima del bus senza diramazione	1000 m (15 m con diramazione)	15 m	1000 m (15 m con diramazione)	15 m
Dimensioni messaggio	Modbus seriale: ● RTU: 256 byte (252 byte di dati) ● ASCII: 513 byte (2x252 byte di dati)	Modbus seriale: ● RTU: 256 byte (252 byte di dati) ● ASCII: 513 byte (2x252 byte di dati)	1024 byte	1024 byte
Utilità	Parole da leggere/bit. Parole da scrivere/bit. Diagnostica.	Parole da leggere/bit. Parole da scrivere/bit. Diagnostica.	Inviare stringa di caratteri Ricevere stringa di caratteri.	Inviare stringa di caratteri Ricevere stringa di caratteri.

Sezione 2.2
Module BMX NOM 0200 a 2 porte RS-485/232

Oggetto della sezione

Questa sezione presenta le comunicazioni seriali sul modulo BMX NOM 0200.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Presentazione del modulo BMX NOM 0200 con 2 porte RS-485/232	26
Apparecchiatura Modicon M340H (Hardened)	32
Messa a terra dei moduli installati	33
Installazione del modulo BMX NOM 0200	35
Considerazioni sul cablaggio del modulo BMX NOM 0200	37

Presentazione del modulo BMX NOM 0200 con 2 porte RS-485/232

Informazioni generali

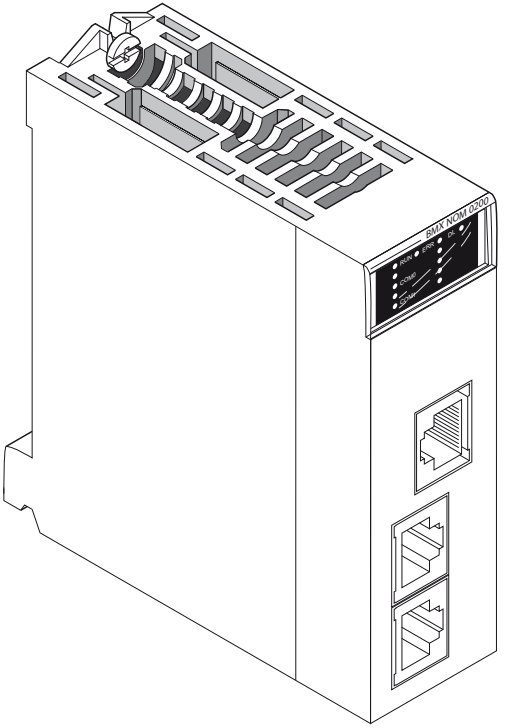
I moduli per collegamento seriale BMX NOM 0200 e BMX NOM 0200H *(vedi pagina 32)* sono 2 moduli asincroni bidirezionali per linea seriale che supportano le comunicazioni Modbus seriale (master o slave) e Modalità caratteri.

Il BMX NOM 0200 è un modulo dedicato in formato semplice che può essere installato su un rack di una stazione Modicon M340.

NOTA: Con temperature estreme (da -25 a 0ºC e da 60 a 70ºC) (-13...32ºF) e (140...158ºF), le caratteristiche di funzionamento del BMX NOM 0200H sono le stesse del BMX NOM 0200 all'interno del relativo campo di temperatura (da 0 a 60 ºC) (32...140ºF).

Introduzione al modulo

L'illustrazione di seguito mostra le caratteristiche fisiche del modulo BMX NOM 0200:



Il modulo BMX NOM 0200 è composto dai seguenti elementi:

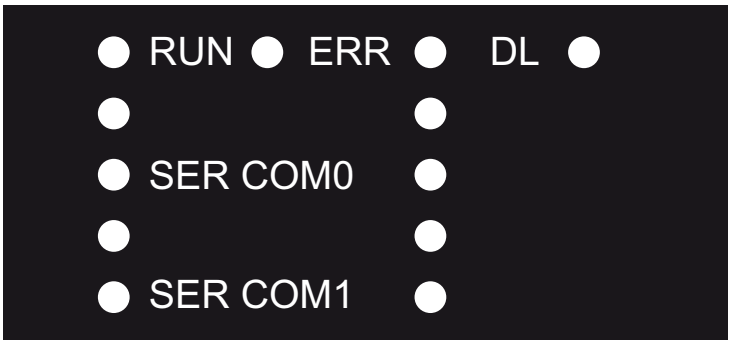
Legenda	Descrizione
1	Cinque indicatori LED sulla parte anteriore del modulo: ● RUN e ERR indicano lo stato del modulo, ● SER COM0 visualizza lo stato del traffico sulla porta 0 o 1 (canale 0), ● SER COM1 visualizza lo stato del traffico sulla porta 2 (canale 1), ● DL indica lo stato del download del firmware.
2	Canale integrato dedicato (canale 0) al collegamento seriale con 2 porte seriali: RS232 (porta 0) e RS485 (porta 1). **Note:** Solo una porta alla volta può essere attiva.
3	Canale integrato dedicato (canale 1) al collegamento seriale con 1 porta seriale: RS485 (porta 2).

NOTA: In alcune modalità operative, **i LED possono indicare informazioni più specifiche** *(vedi pagina 27)*.

Diagnostica visiva

Cinque LED sono posizionati sul pannello anteriore del modulo BMX NOM 0200. Essi forniscono informazioni relative allo stato di funzionamento e di comunicazione del modulo del collegamento seriale integrato.

Display a LED:



- RUN = Il modulo è alimentato e ben configurato.
- ERR = Il modulo ha rilevato un errore e non può funzionare correttamente.
- DL = Il download del firmware è in corso.
- SER COM0 = Comunicazione rilevata sulla porta 0 o 1 (canale 0).
- SER COM1 = Comunicazione rilevata sulla porta 2 (canale 1).

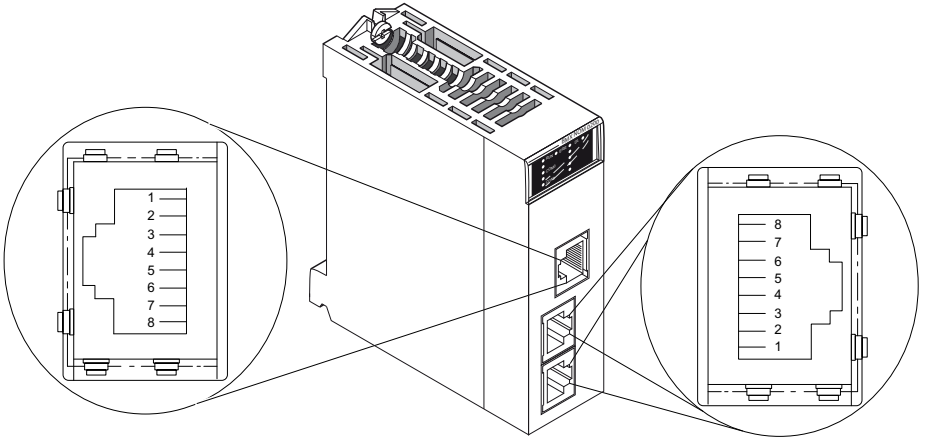
Significato dei LED:

- Ciascun LED può essere in uno di questi stati:
  - 1 = Acceso
  - 0 = Spento
  - B = Lampeggiante
- All'avvio del modulo, tutti i LED vengono accesi e quindi spenti, allo scopo di verificarne il funzionamento corretto.

RUN	ERR	SER COM0	SER COM1	DL	Diagnosi
0	_	_	_	_	Il modulo non è alimentato o non è in funzione.
0	B	_	_	_	Il modulo non è configurato.
1	1	_	_	_	Il modulo funziona in modo anomalo.
1	1	1	0	_	Il modulo ha rilevato un problema sul canale 0.
1	1	1	B	_	Il modulo ha rilevato un problema sul canale 0, il canale 1 sta scambiando dati.
1	1	0	1	_	Il modulo ha rilevato un problema sul canale 1.
1	1	B	1	_	Il modulo ha rilevato un problema sul canale 1, il canale 0 sta scambiando dati.
1	0	B	_	_	Il canale 0 sta scambiando dati.
1	0	_	B	_	Il canale 1 sta scambiando dati.
B	B	_	_	0	Manca la CPU.
B	B	B	B	_	Il modulo sta eseguendo i test automatici.
_	_	_	_	B	Il download di un firmware del modulo è in corso.
_	_	_	_	1	Il firmware viene caricato; è necessario reimpostare il modulo.

Introduzione alle porte seriali

L'illustrazione di seguito mostra le porte seriali RJ45 sul BMX NOM 0200:



La tabella di seguito mostra l'assegnazione dei pin per la porta seriale sul BMX NOM 0200:

N° pin	RS485 canale 1 / porta 1 o 2	RS232 canale 0 / porta 0
1	_	RXD (Ricezione dati)
2	_	TXD (Trasmissione dati)
3	_	RTS (Richiesta di invio)
4	D1 (B/B4)	DTR (Terminale dati pronto)
5	D0 (A/A4)	DSR (Data Set Ready)
6	_	CTS (Clear To Send)
7	_	DCD (Data Carrier Detect)
8	messa a terra collegamento seriale potenziale (0 V)	messa a terra collegamento seriale potenziale (0 V)

NOTA:

- Le due linee RS485 sono isolate. La tensione di isolamento tra le due linee seriali è 500 V e anche tra ogni linea seriale isolata e il backplane la tensione di isolamento può arrivare fino a 500V AC.
- L'RS232 a sette fili e l'RS485 a due fili utilizzano lo stesso connettore femmina RJ45. Solo il cablaggio di segnale è diverso.

Specifiche dei canali

I canali del modulo BMX NOM 0200 comprendono:

- Due interfacce fisiche isolate RS485,
- Un'interfaccia fisica RS232 non isolata,
- Tipi di comunicazione Modbus Seriale (ASCII e RTU) e Modalità caratteri.

Le specifiche di collegamento per i due protocolli sono:

	Modbus seriale / RS485	**Modbus Seriale / RS232**	**Modalità caratteri / RS485**	**Modalità caratteri / RS232**
Tipo	Master/Slave	Master/Slave	Half Duplex	Full Duplex
Flow	19200 baud. I parametri possono essere impostati da 300 baud a 57600 baud.	19200 baud. I parametri possono essere impostati da 300 baud a 115200 baud.	9600 baud. I parametri possono essere impostati da 300 baud a 57600 baud.	9600 baud. I parametri possono essere impostati da 300 baud a 115200 baud.
Numero di apparecchiature	32	32	_	_
Indirizzi slave autorizzati	da 1 a 247	da 1 a 247	_	_
Lunghezza massima del bus senza diramazione	Fare riferimento alla tabella di sotto (15 m con derivazioni)	15 m	Fare riferimento alla tabella di sotto (15 m con derivazioni)	15 m
Dimensioni messaggio	Modbus seriale: ● RTU: 256 byte (252 byte di dati) ● ASCII: 513 byte (2x252 byte di dati)	Modbus seriale: ● RTU: 256 byte (252 byte di dati) ● ASCII: 513 byte (2x252 byte di dati)	1024 byte	1024 byte
Utilità	Parole da leggere/bit. Parole da scrivere/bit. Diagnostica.	Parole da leggere/bit. Parole da scrivere/bit. Diagnostica.	Inviare stringa di caratteri Ricevere stringa di caratteri.	Inviare stringa di caratteri Ricevere stringa di caratteri.
Controllo flusso hardware	_	Opzionalmente tramite i segnali RTS/CTS.	_	Opzionalmente tramite i segnali RTS/CTS.

La tabella di sotto mostra la lunghezza massima del cavo RS485 che può essere usata, in base alla velocità di trasmissione scelta:

Velocità di trasmissione scelta (bit/s)	Lunghezza (m)	Codice prodotto
300	1000	(1)
600	1000	(1)
1200	1000	(1)
2400	1000	(1)
9600	1000	(1)
19200	600	(1)
38400	300	(1) o (2)
57600	200	(1) o (2)

- (1): Cavo schermato a coppia intrecciata, sezione AWG24 (TSX CSA 100, TSX CSA 200, TSX CSA 500)
- (2): Cavo di categoria 5 o più alto

Consumo del modulo BMX NOM 0200

Questa tabella mostra il consumo del modulo BMX NOM 0200:

Tensione	Corrente tipica	Corrente massima	Dissipazione di potenza tipica	Dissipazione di potenza massima
24 V CC	80 mA	130 mA	1,92 W	3,12 W

Apparecchiatura Modicon M340H (Hardened)

M340H

L'apparecchiatura Modicon M340H (hardened) è una versione rinforzata del modello M340. Le sue caratteristiche ne permettono l'uso anche a temperature estreme (-25 ... 70 ºC) (-13 ... 158 ºF) e in ambienti chimici aggressivi.

Questo trattamento aumenta le capacità di isolamento dei circuiti stampati e la loro resistenza a:
- condensa
- polveri (particelle estranee)
- corrosione chimica, in particolare per l'uso in ambienti sulfurei (impianti petroliferi, di purificazione, raffinerie ecc.) o atmosfere contenenti alogeni (cloro, ecc.)

Quando è utilizzata nel campo di temperatura standard (0 … 60 ºC) (32 ... 140 ºF), l'apparecchiatura M340H ha le stesse caratteristiche prestazionali dell'M340 standard.

A temperature estreme (-25... 0ºC and 60... 70ºC) (-13...32ºF and 140...158ºF), le versioni Hardened possono avere valori nominali di potenza ridotti che influenzano il calcolo della potenza per le applicazioni Unity Pro.

Se questa apparecchiatura viene utilizzata a valori di temperatura esterni al campo -25...70 ºC (-13...158 ºF), possono verificarsi delle anomalie.

⚠ ATTENZIONE

FUNZIONAMENTO ANOMALO DELL'APPARECCHIATURA

Non utilizzare l'apparecchiatura M340H al di fuori del campo di temperatura specificato.

Il mancato rispetto di queste istruzioni può provocare infortuni o danni alle apparecchiature.

L'apparecchiatura Hardened è dotata di un rivestimento conforme per la protezione delle schede elettroniche. Questa protezione, se associata a condizioni adeguate di installazione e manutenzione, aumenta la robustezza per l'esercizio in ambienti chimici aggressivi.

Messa a terra dei moduli installati

Informazioni generali

La messa a terra dei moduli Modicon M340 è fondamentale per evitare le scosse elettriche.

Messa a terra dei processori e degli alimentatori

⚠ ⚠ PERICOLO

RISCHIO DI SHOCK ELETTRICO, ESPLOSIONE O ARCO ELETTRICO

Assicurarsi che i contatti di messa a terra sia presenti e non deformati. Se alcuni lo sono, non usare il modulo e rivolgersi al rappresentante Schneider Electric.

Il mancato rispetto di queste istruzioni provocherà morte o gravi infortuni.

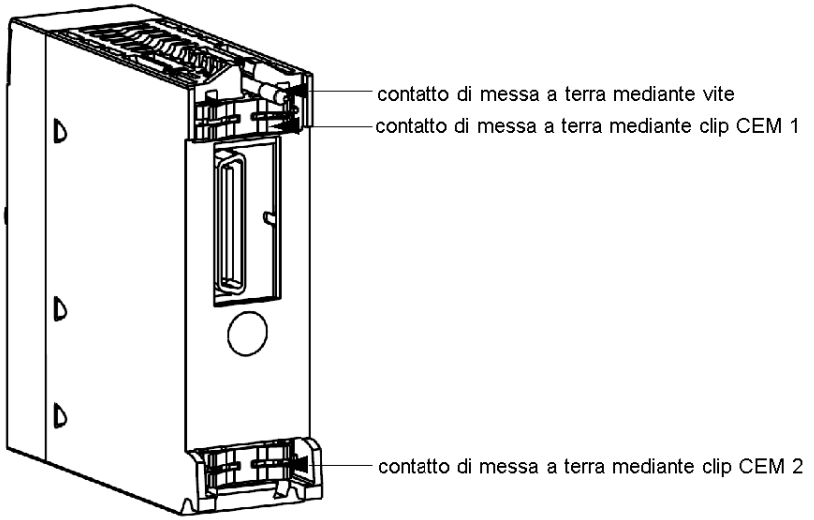
⚠ AVVERTENZA

FUNZIONAMENTO ANOMALO DELL'APPARECCHIATURA

Serrare le viti di montaggio dei moduli. Se il circuito venisse interrotto, potrebbe verificarsi un funzionamento anomalo del sistema.

Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.

Tutti i moduli Modicon M340 sono dotati di contatti di messa a terra sul retro:


contatto di messa a terra mediante vite
contatto di messa a terra mediante clip CEM 1
contatto di messa a terra mediante clip CEM 2

questi contatti permettono di collegare il bus di messa a terra al bus di messa a terra del rack.

Installazione del modulo BMX NOM 0200

Informazioni generali

Il modulo BMX NOM 0200 viene installato in un rack della stazione Modicon M340 e non può utilizzare gli slot necessari per l'alimentazione e il processore. Questa installazione deve conformarsi alle istruzioni di installazione del rack.

Il modulo BMX NOM 0200 richiede l'installazione di una CPU con versione minima del SO 02.10. L'installazione deve avvenire conformemente alle istruzioni di installazione della CPU.

Un connettore RJ45 può quindi essere collegato al modulo in base alla rete di destinazione.

NOTA: Il modulo BMX NOM 0200 può essere installato in un rack mentre l'applicazione viene eseguita sul PLC.

⚠ AVVERTENZA

FUNZIONAMENTO ANOMALO DELL'APPARECCHIATURA

L'applicazione di questo prodotto richiede esperienza di progettazione e programmazione dei sistemi di controllo. Solo il personale in possesso di tali competenze è autorizzato a programmare, installare, modificare e utilizzare questi prodotti.

Rispettare la regolamentazione e tutte le norme locali e nazionali sulla sicurezza.

Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.

Numero di moduli

Siccome il numero dei canali esperti gestiti da una stazione PLC è collegato al processore installato, pertanto il numero massimo dei moduli BMX NOM 0200 in una stazione dipenderà da:

- Il numero di canali configurati su ciascun modulo BMX NOM 200 (ciascun canale conta come canale esperto),
- Il tipo e versione di processore installato *(vedi Modicon M340 con Unity Pro, Processori, rack e moduli alimentatori, Manuale di configurazione)*,
- Il numero di canali esperti già in uso.

Alla creazione dell'applicazione, Unity Pro permette di verificare che la limitazione non venga superata.

Connessione/Disconnessione

Il modulo BMX NOM 0200 può essere collegato o scollegato con le apparecchiature sotto tensione. Quando il modulo viene scollegato dal rack, la sua memoria interna viene cancellata. Il modulo attraversa una fase di inizializzazione dopo che viene ricollegato al backplane.

Un NOM0200 (dalla versione V1.2) può essere inserito in un rack in qualsiasi slot libero senza essere stato configurato. Questa possibilità è vantaggiosa per collegare una PUNIT quando la CPU non è configurata o come un punto di collegamento aggiuntivo. In questo caso il BMX NOM0200 si trova nella configurazione predefinita.

La configurazione predefinita del BMX NOM 0200 è slave MODBUS all'indirizzo 248, RTU (ritardo tra i frame = 2ms), 8 bit di dati, 1 bit di stop, parità pari, RS232 a 115200bit/s sul canale 0 e RS485 a 57600bit/s sul canale 1.

L'indirizzo 248 è l'indirizzo punto-punto al quale qualsiasi modulo slave BMX NOM 0200 risponde. Questa funzionalità consente di collegarsi direttamente a qualsiasi modulo slave il cui indirizzo è sconosciuto.

Aggiornamento firmware

Il firmware BMX NOM 0200 può essere aggiornato tramite il backplane del PLC. L'aggiornamento del firmware è descritto nel documento *Unity Loader, a SoCollaborative software - Guida utente.*

Considerazioni sul cablaggio del modulo BMX NOM 0200

Considerazioni operative

⚠ AVVERTENZA

FUNZIONAMENTO ANOMALO DELL'APPARECCHIATURA

Benché sia possibile collegare e scollegare i cavi del modulo BMX NOM 0200 e delle CPU BMX P34 20x0 con la stazione BMX XBP sotto tensione, queste operazioni possono interrompere l'esecuzione in corso dell'applicazione.

Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.

Collegamento

Le situazioni descritte di seguito possono interrompere temporaneamente l'applicazione o le comunicazioni:

- Il connettore RJ45 viene collegato o scollegato mentre l'apparecchiatura è sotto tensione.
- I moduli vengono reinizializzati mentre si ripristina l'alimentazione.

Capitolo 3
Architetture di comunicazione seriale

Oggetto del capitolo

Questo capitolo presenta un'introduzione alle architetture che utilizzano la comunicazione seriale sui processori BMX P34 1000/2000/2010/20102/2020 e sul modulo BMX NOM 0200, oltre al cablaggio richiesto.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
3.1	Architettura delle comunicazioni seriali per i processori BMX P34 1000/2000/2010/20102/2020	40
3.2	Architetture di comunicazione seriale per BMX NOM 0200	50
3.3	Cablaggio	60

Sezione 3.1
Architettura delle comunicazioni seriali per i processori BMX P34 1000/2000/2010/20102/2020

Oggetto della sezione

Questa sezione presenta un'introduzione alle architetture che utilizzano la comunicazione seriale sui processori BMX P34 11000/2000/2010/20102/2020, oltre ai requisiti di cablaggio.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Terminazione della linea Modbus e polarizzazione (RS485)	41
Collegamento dei dispositivi Modbus (RS485)	43
Connessione di un'apparecchiatura per terminale dati (DTE) (RS232)	46
Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)	48

Terminazione della linea Modbus e polarizzazione (RS485)

Panoramica

Una rete Modbus multipunto deve avere una terminazione di linea e una polarizzazione.

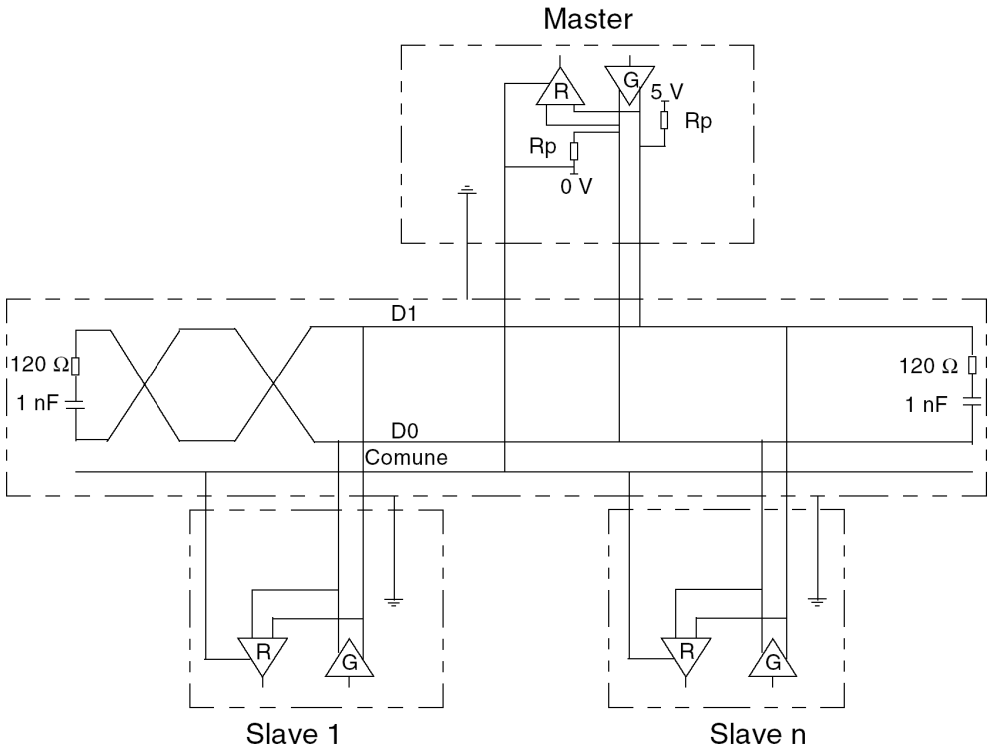
Le apparecchiature collegabili a questo bus sono:

- Altri PLC come M340, Premium, Quantum, Twido o Nano
- Dispositivi di automazione Schneider come Altivar, Modulo di sicurezza XPS, SEPAM, XBT o Momentum
- Altri dispositivi conformi al protocollo Modbus
- Modem, Hub

Un esempio della rete **Modbus multipunto** *(vedi pagina 45)* compreso un processore BMX P34 2010 viene presentato in questo manuale.

NOTA: Può anche essere eseguita una rete Modbus punto-punto

Schema elettrico della terminazione di linea e della polarizzazione:



Terminazione di linea

La terminazione di linea viene eseguita esternamente: è costituito da due resistenze da 120 Ω e un condensatore 1 nF, posizionati ad ogni estremità della rete (VW3 A8 306 RC or VW3 A8 306 DRC).

Non posizionare la terminazione di linea al capo di un cavo di derivazione.

Polarizzazione della linea

Su una linea Modbus, è necessaria la polarizzazione per una rete RS485.

- Se la CPU M340 CPU viene usata come master, **viene automaticamente azionata dal sistema** *(vedi pagina 23)* pertanto non è necessaria alcuna polarizzazione esterna.
- Se la CPU M340 viene usata come slave, la polarizzazione deve essere implementata da due resistenze da 450-650 Ω (Rp) collegate sulla coppia equilibrata RS485:
  - una resistenza pull-up fino a una tensione di 5 V sul circuito D1,
  - una resistenza pull-down al circuito comune sul circuito D0.

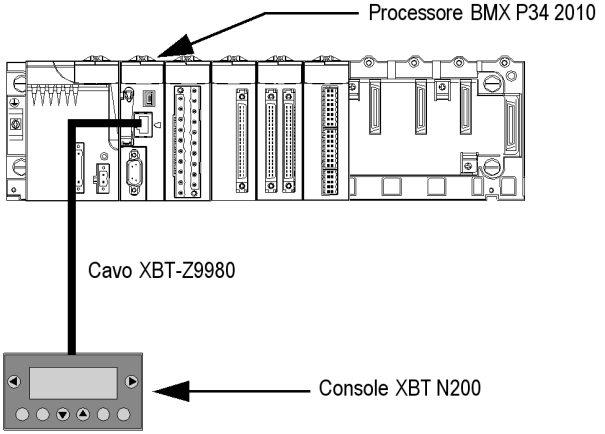
Collegamento dei dispositivi Modbus (RS485)

Informazioni generali

Nelle pagine seguenti vengono presentati due esempi di collegamento di dispositivi Modbus e di un'architettura di collegamento seriale Modbus.

Collegamento di dispositivi Modbus alimentati tramite collegamento seriale

La seguente figura mostra il collegamento di un processore BMX P34 2010 ad una console XBT N200 alimentata tramite il link seriale Modbus:



I dispositivi sono configurati nel modo seguente:

- Il processore BMX P34 2010 è configurato come slave,
- interfaccia uomo macchina XBTN200 configurata come master.

Il cavo XBT-Z9980 presenta le seguenti proprietà:

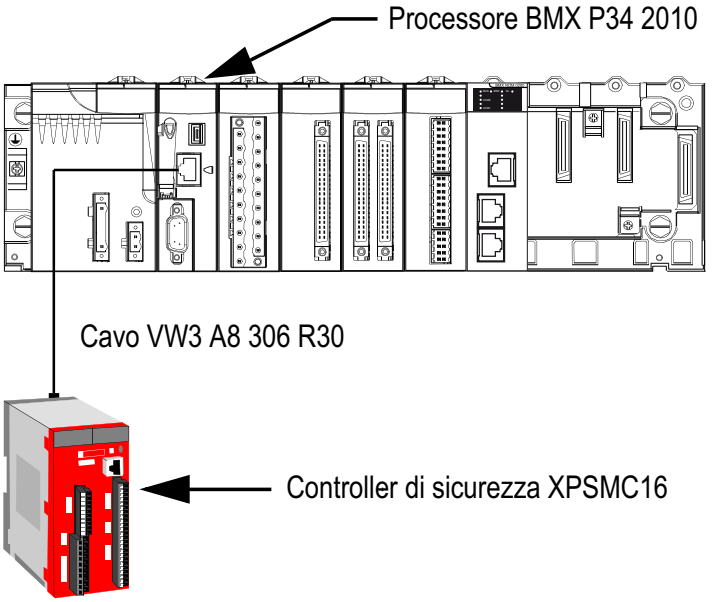
- Connessione: 2 connettori maschio RJ45
- Cablaggio: 2 cavi per la linea fisica RS485 e 2 per l'alimentazione da collegamento seriale

Collegamento di dispositivi Modbus non alimentati tramite collegamento seriale

L'architettura comprende gli elementi seguenti:

- Un processore BMX P34 2010,
- Un controller di sicurezza XPSMC16.

L'illustrazione seguente mostra la connessione di un processore BMX P34 2010 a un controller di sicurezza XPSMC16:



I dispositivi sono configurati nel modo seguente:

- Il processore BMX P34 2010 è configurato come master
- Il controller di sicurezza XPSMC16 è configurato come slave.

Il cavo VW3 A8306 R30 presenta le seguenti proprietà:

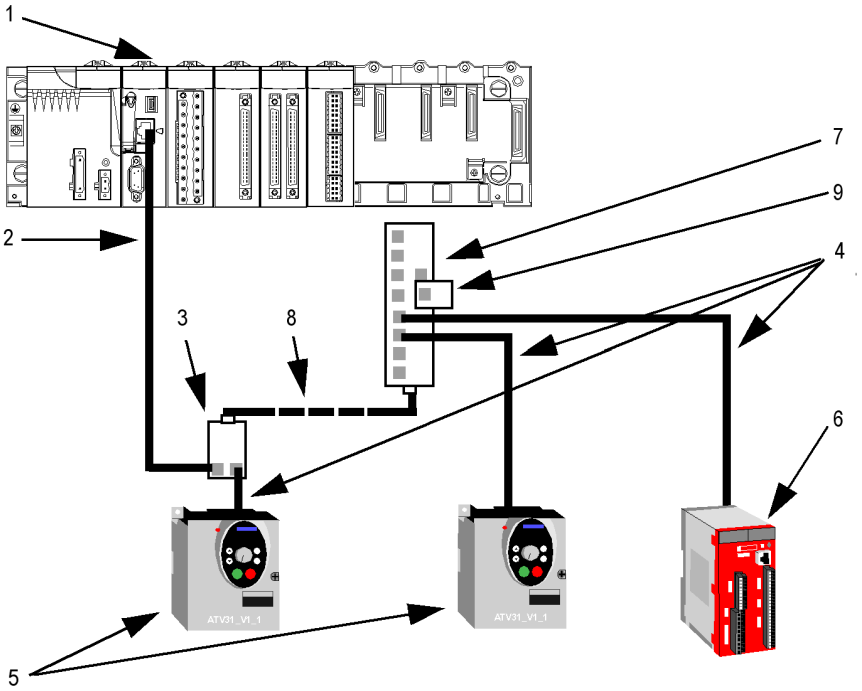
- Connessione: 2 connettori maschio RJ45
- Cablaggio: 2 fili per la linea fisica RS485

Architettura collegamento seriale Modbus

L'architettura del collegamento seriale Modbus comprende gli elementi seguenti:

- Un processore BMX P34 2010/20102 configurato come master,
- Un controller di sicurezza XPSMC16 configurato come slave
- un blocco di connessione splitter isolato TWDXCAISO
- un blocco di connessione splitter LU9 GC3
- due unità ATV31, configurate come slave.

Lo schema seguente mostra l'architettura del collegamento seriale descritta sopra:



1 processore BMX P34 2010
2 cavo XBT-Z9980
3 un blocco splitter isolato TWDXCAISO
4 cavo VW3 A8 306 R30
5 azionamento ATV31
6 controller di sicurezza XPSMC16
7 blocco splitter LU9GC3
8 cavo TSXCSAx00
9 cavo VW3 A8 306 R

Connessione di un’apparecchiatura per terminale dati (DTE) (RS232)

Generale

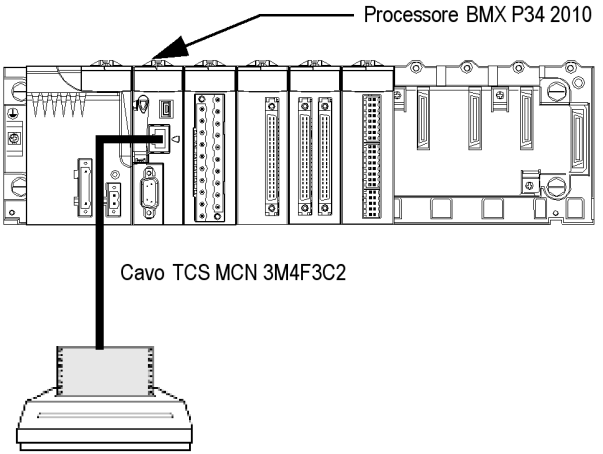
Apparecchiatura terminale dati è il termine utilizzato per descrivere dispositivi quali:

- periferiche comuni (stampanti, schermi tattili, terminali di laboratorio, ecc.),
- periferiche specializzate (lettori di codice a barre, ecc.),
- PC.

Tutte le apparecchiature terminale dati sono collegate a un processore BMX P34 1000/2000/2010/20102/2020 tramite un cavo incrociato seriale mediante il collegamento fisico RS232.

Connessione apparecchiatura terminale dati

L’illustrazione seguente mostra il collegamento di una stampante con un processore BMX P34 2010:



Il protocollo di comunicazione utilizzato è la modalità carattere.

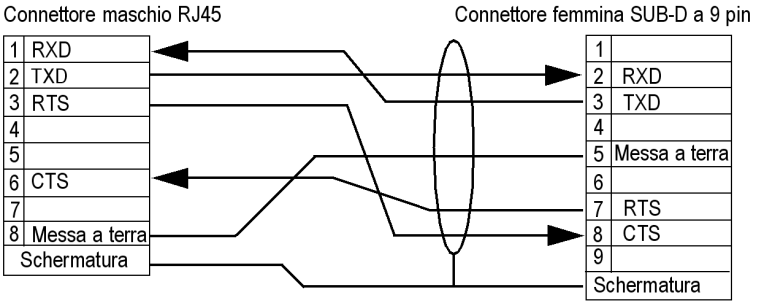
NOTA: è possibile collegare solo un’apparecchiatura terminale dati a ciascun processore BMX P34 1000/2000/2010/20102/2020.

Cavo incrociato seriale RS 232

Il cavo incrociato seriale TCS MCN 3M4F3C2 presenta due connettori:

- maschio RJ45
- femmina SUB-D a 9 pin

L'illustrazione seguente mostra l'assegnazione dei pin di un cavo incrociato seriale TCS MCN 3M4F3C2:



Accessori e cavi di collegamento

La tabella seguente mostra i codici prodotto di cavi e schede da utilizzare in base al connettore seriale utilizzato dall'apparecchiatura terminale dati:

Connettore seriale per apparecchiatura terminale dati	**Cablaggio**
Connettore maschio SUB-D a 9 pin	Cavo TCS MCN 3M4F3C2
Connettore maschio SUB-D a 25 pin	• Cavo TCS MCN 3M4F3C2 • Scheda TSX CTC 07
Connettore femmina SUB-D a 25 pin	• Cavo TCS MCN 3M4F3C2 • Scheda TSX CTC 10

Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)

Generale

Apparecchiatura di terminazione circuito dati (DCE) è il termine utilizzato per descrivere dispositivi come i modem.

Per un dispositivo di tipo DCE, i pin RTS e CTS sono collegati direttamente (non incrociati).

Tutte le apparecchiature di terminazione circuito dati sono collegate a un processore BMX P34 1000/2000/2010/20102/2020 tramite un cavo diretto seriale mediante il collegamento fisico RS232.

NOTA: Le differenze tra le connessioni DCE e DTE sono in buona parte nelle spine e nella direzione del segnale dei pin (ingresso o uscita). Per esempio, un PC desktop viene definito come un dispositivo DTE mentre un modem viene definito come un dispositivo DCE.

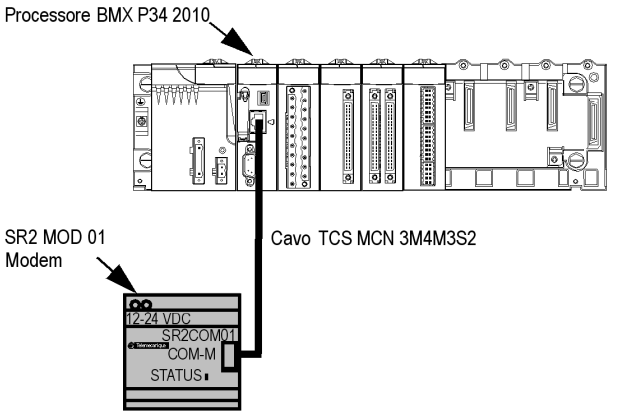
Caratteristiche del modem

le CPU M340 funzionano con la maggior parte dei modem sul mercato. Per collegare un modem alla porta seriale di un processore BMX P34 1000/2000/2010/20102/2020, il modem deve presentare le caratteristiche seguenti:

- Supportare 10 o 11 bit per carattere se la porta del terminale viene usato in Modbus Seriale:
  - 7 o 8 bit di dati
  - 1 o 2 bit di stop
  - parità pari, dispari o nessuna
- funzionamento senza controllo portante dati

Connessione apparecchiatura di terminazione circuito dati

L'illustrazione seguente mostra il collegamento di un modem con un processore BMX P34 2010:



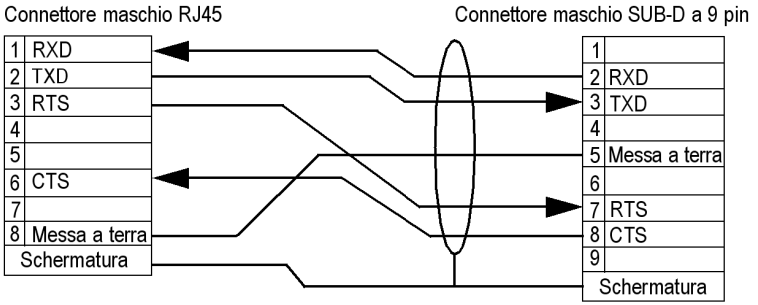
NOTA: In Modbus Seriale, il tempo di attesa deve essere compreso tra 100 e 250 ms.

Cavo diretto seriale RS 232

Il cavo diretto seriale TCS MCN 3M4M3S2 presenta due connettori:

- maschio RJ45,
- maschio SUB-D a 9 pin.

L'illustrazione seguente mostra l'assegnazione dei pin di un cavo diretto seriale TCS MCN 3M4M3S2 :



Accessori e cavi di collegamento

La tabella seguente mostra i codici prodotto di cavi e adattatori da utilizzare in base al connettore seriale utilizzato dall'apparecchiatura di terminazione circuito dati:

Connettore seriale per apparecchiatura di terminazione circuito dati	**Cablaggio**
Connettore femmina SUB-D a 9 pin	Cavo TCS MCN 3M4M3S2
Connettore femmina SUB-D a 25 pin	• Cavo TCS MCN 3M4M3S2 • Scheda TSX CTC 09

Sezione 3.2
Architetture di comunicazione seriale per BMX NOM 0200

Oggetto della sezione

Questa sezione presenta un'introduzione alle architetture che utilizzano la comunicazione seriale sul modulo BMX NOM 0200, oltre ai requisiti di cablaggio.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Terminazione della linea Modbus e polarizzazione (RS485)	51
Collegamento dei dispositivi Modbus (RS485)	53
Connessione di un'apparecchiatura per terminale dati (DTE) (RS232)	55
Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)	57

Terminazione della linea Modbus e polarizzazione (RS485)

Panoramica

Una rete Modbus multipunto deve avere una terminazione di linea e una polarizzazione.

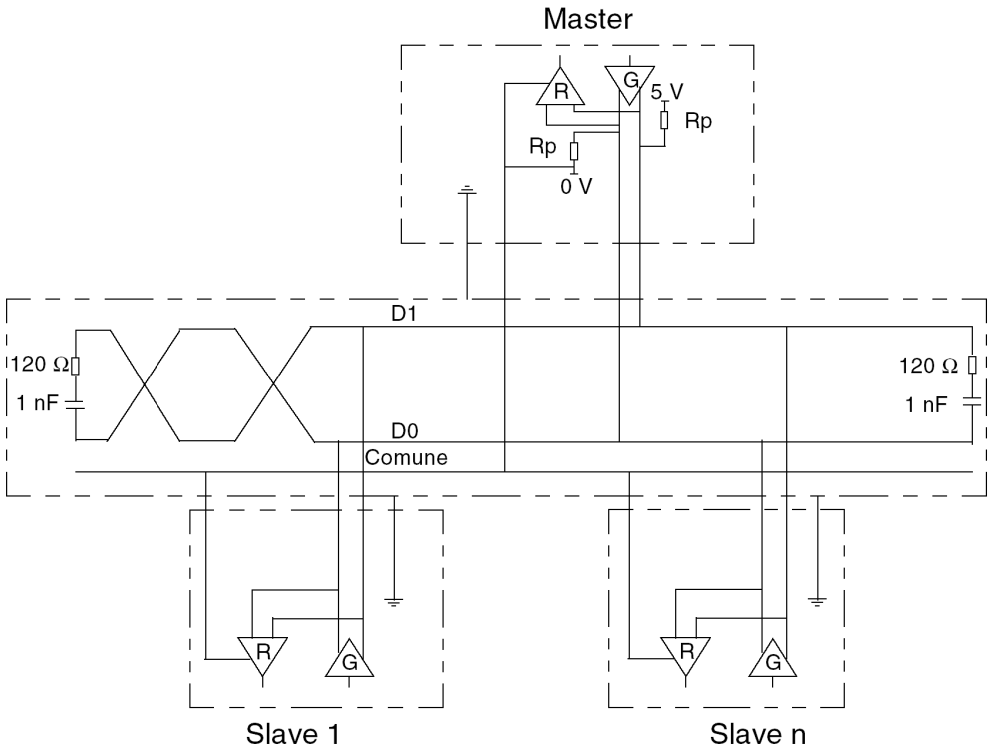
Gli apparecchi collegabili a questo bus sono:

- Altri PLC come M340, Premium, Quantum, Twido o Nano
- Dispositivi di automazione Schneider come Altivar, Modulo di sicurezza XPS, SEPAM, XBT o Momentum
- Altri dispositivi conformi al protocollo Modbus
- Modem, Hub

Un esempio della rete **Modbus multipunto** *(vedi pagina 54)* compreso un modulo BMX NOM 0200 viene presentato in questo manuale.

NOTA: Può anche essere eseguita una rete Modbus punto-punto

Schema elettrico della terminazione di linea e della polarizzazione:



Terminazione di linea

La terminazione di linea viene eseguita esternamente: è costituito da due resistenze da 120 Ω e da un condensatore di 1 nF, posizionati ad ogni estremità della rete (VW3 A8 306RC o VW3 A8 306 DRC). Non posizionare la terminazione di linea al capo di un cavo di derivazione.

Polarizzazione della linea

Su una linea Modbus, è necessaria la polarizzazione per una rete RS485.

- Se il modulo BMX NOM 0200 viene usato come master, viene automaticamente azionato dal sistema pertanto non è necessaria alcuna polarizzazione esterna.
- Se il modulo BMX NOM 0200 viene usato come slave, la polarizzazione deve essere implementata da due resistenze da 450-650 Ω (Rp) collegate sulla coppia equilibrata RS485:
  - una resistenza pull-up fino a una tensione di 5 V sul circuito D1,
  - una resistenza pull-down al circuito comune sul circuito D0.

NOTA:

In modalità carattere, la polarizzazione della linea è configurabile in Unity Pro. È possibile scegliere tra:

- polarizzazione a bassa impedenza come nelle reti Modbus (lo scopo di questo tipo di polarizzazione è di consentire al master di mantenere lo stato predefinito),
- polarizzazione ad alta impedenza (lo scopo di questo tipo di polarizzazione è di consentire a ciascun dispositivo di mantenere lo stato predefinito),
- nessuna polarizzazione (se viene usata una polarizzazione esterna).

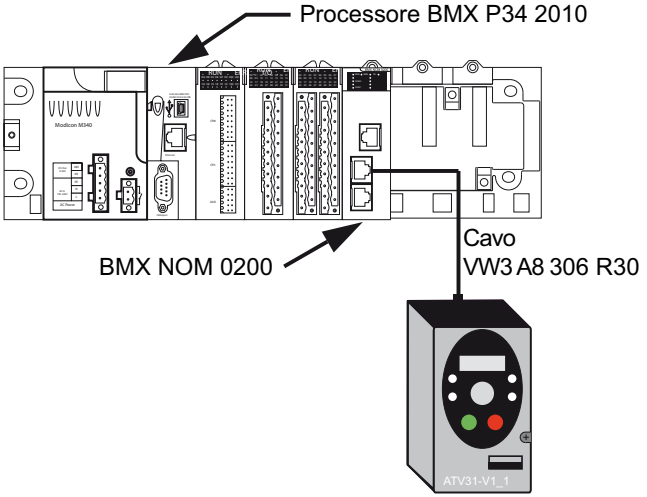
Collegamento dei dispositivi Modbus (RS485)

Informazioni generali

Nelle pagine seguenti viene presentato un esempio di collegamento di dispositivo Modbus e di un'architettura di collegamento seriale Modbus.

Collegamento di dispositivi Modbus non alimentati tramite collegamento seriale

La figura seguente mostra un modulo BMX NOM 0200 collegato a un azionamento ATV31.



I dispositivi sono configurati nel modo seguente:

- un processore BMX P34 2010
- un modulo BMX NOM 0200 configurato come master
- Un azionamento ATV31 configurato come slave.

Il cavo VW3 A8306 R30 presenta le seguenti proprietà:

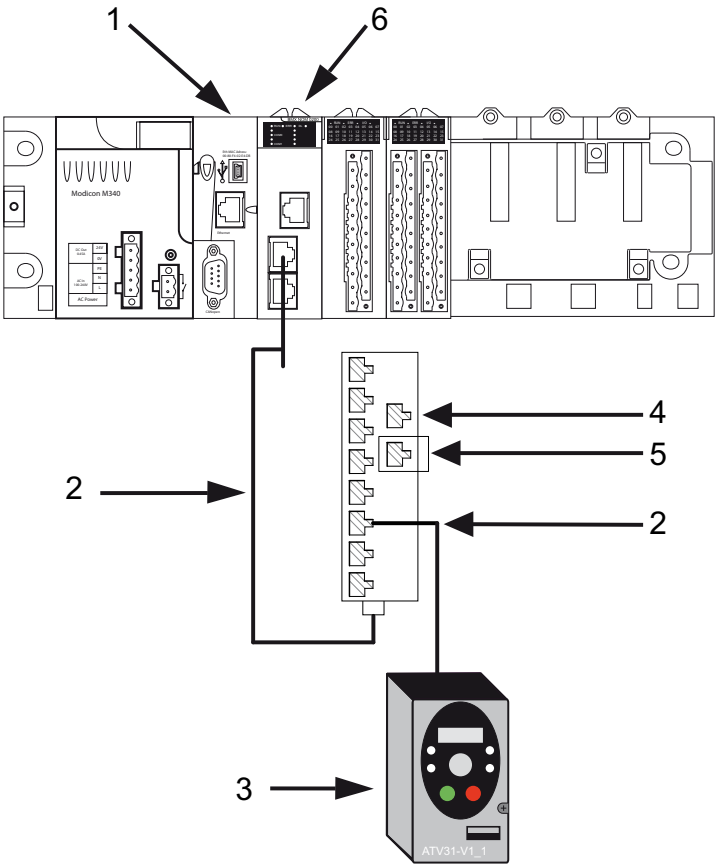
- Connessione: 2 connettori maschio RJ45
- Cablaggio: 2 fili per la linea fisica RS485

Architettura di collegamento seriale Modbus

L'architettura del collegamento seriale Modbus comprende gli elementi seguenti:

- un processore BMX P34 2010
- un modulo BMX NOM 0200 configurato come master
- un blocco di connessione splitter isolato TWDXCAISO
- un blocco di connessione splitter LU9 GC3
- Due ATV31, configurati come slave.

L'illustrazione seguente mostra l'architettura del collegamento seriale descritta sopra:



1 processore BMX P34 2010
2 cavo VW3 A8 306 R30
3 azionamento ATV31
4 blocco di connessione splitter LU9GC3
5 cavo VW3 A8 306 R
6 modulo BMX NOM 0200

Connessione di un’apparecchiatura per terminale dati (DTE) (RS232)

Generale

Apparecchiatura terminale dati è il termine utilizzato per descrivere dispositivi quali:

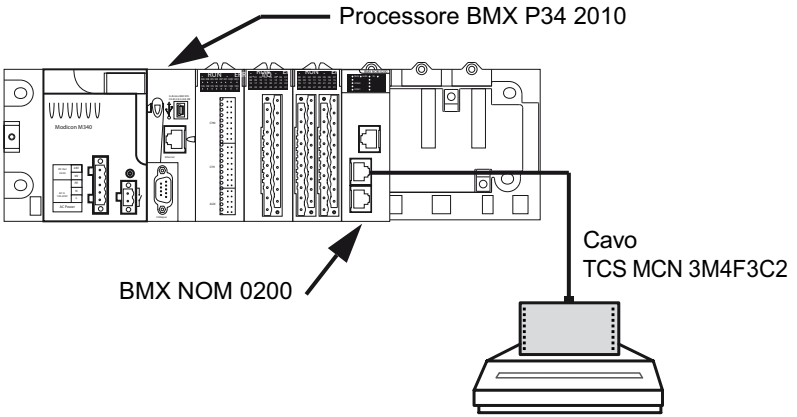
- periferiche comuni (stampanti, schermi tattili, terminali di laboratorio, ecc.);
- periferiche specializzate (lettori di codice a barre, ecc.);
- PC.

Per un dispositivo di tipo DCE, i pin RTS e CTS sono incrociati.

Tutte le apparecchiature terminale dati sono collegate a un modulo BMX NOM 0200 tramite un cavo incrociato seriale mediante il collegamento fisico RS232.

Connessione apparecchiatura terminale dati

La figura di seguito mostra una stampante collegata al modulo BMX NOM 0200:



Il protocollo di comunicazione utilizzato è la modalità carattere.

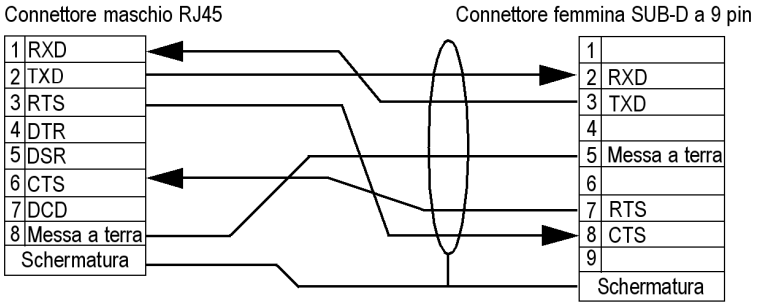
NOTA: È possibile collegare solo un’apparecchiatura terminale dati al modulo BMX NOM 0200.

Cavo incrociato seriale RS 232

Il cavo incrociato seriale TCS MCN 3M4F3C2 presenta due connettori:

- maschio RJ45,
- femmina SUB-D a 9 pin.

La figura seguente mostra l'assegnazione dei pin di un cavo incrociato seriale TCS MCN 3M4F3C2:



Accessori e cavi di collegamento

La tabella seguente mostra i codici prodotto di cavi e schede da utilizzare in base al connettore seriale utilizzato dall'apparecchiatura terminale dati:

Connettore seriale per apparecchiatura terminale dati	**Cablaggio**
Connettore maschio SUB-D a 9 pin	Cavo TCS MCN 3M4F3C2
Connettore maschio SUB-D a 25 pin	• Cavo TCS MCN 3M4F3C2 • Scheda TSX CTC 07
Connettore femmina SUB-D a 25 pin	• Cavo TCS MCN 3M4F3C2 • Scheda TSX CTC 10

Connessione apparecchiatura di terminazione circuito dati (DCE) (RS232)

Generale

Apparecchiatura di terminazione circuito dati (DCE) è il termine utilizzato per descrivere dispositivi come i modem.

Per un dispositivo di tipo DCE, i pin RTS e CTS sono collegati direttamente (non incrociati).

Tutte le apparecchiature di terminazione circuito dati sono collegate a un modulo BMX NOM 0200 tramite un cavo seriale diretto mediante il collegamento fisico RS232.

NOTA: Le differenze tra le connessioni DCE e DTE sono in buona parte nelle spine e nella direzione del segnale dei pin (ingresso o uscita). Per esempio, un PC desktop viene definito come un dispositivo DTE mentre un modem viene definito come un dispositivo DCE.

Caratteristiche del modem

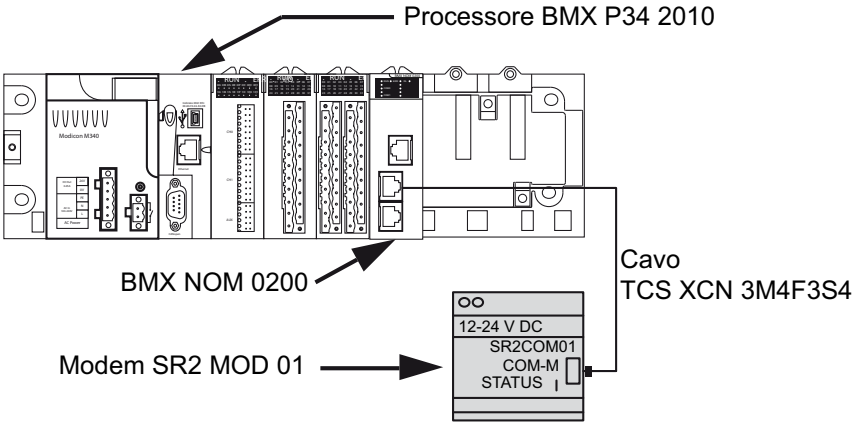
Il modulo BMX NOM 0200 funziona con la maggior parte dei modem sul mercato. Per collegare un modem alla porta seriale di un modulo BMX NOM 0200, il modem deve presentare le caratteristiche seguenti:

- Supportare 10 o 11 bit per carattere se la porta del terminale viene usato in Modbus Seriale:
  - 7 o 8 bit di dati
  - 1 o 2 bit di stop
  - parità pari, dispari o nessuna
- funzionamento senza controllo portante dati

I segnali CTS, DTR, DSR e DCD possono essere gestiti dall’applicazione.

Connessione apparecchiatura di terminazione circuito dati

La figura di seguito mostra un modem collegato al modulo BMX NOM 0200:



Il collegamento del modem richiede il funzionamento di un cavo modem specifico.

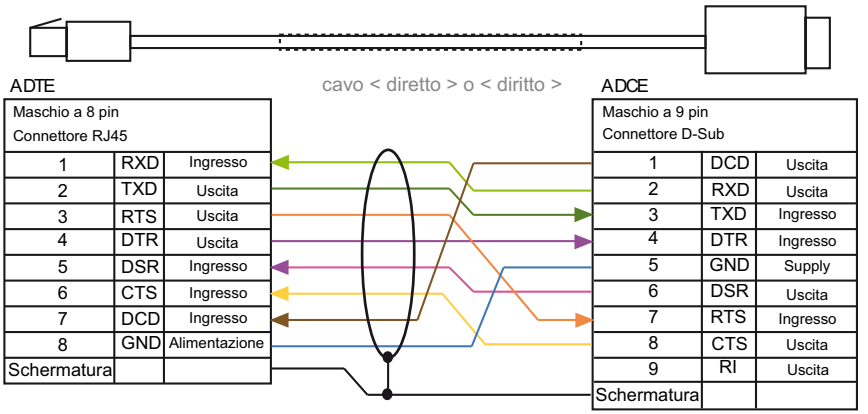
Cavo diretto seriale RS 232

Esempio del cavo TCS XCN 3M4F3S4:

Il cavo diretto seriale TCS XCN 3M4F3S4 è una versione a 8 fili e ha due connettori:

- maschio RJ45,
- maschio SUB-D a 9 pin.

L'illustrazione seguente mostra l'assegnazione dei pin di un cavo diretto seriale TCS XCN 3M4F3S4:



Accessori e cavi di collegamento

La tabella seguente mostra i codici prodotto di cavi e adattatori da utilizzare in base al connettore seriale utilizzato dall'apparecchiatura di terminazione circuito dati:

Connettore seriale per apparecchiatura di terminazione circuito dati	Cablaggio
Connettore femmina SUB-D a 9 pin	• Cavo TCS MCN 3M4M3S2 • Cavo TCS XCN 3M4F3S4
Connettore femmina SUB-D a 25 pin	• Cavo TCS MCN 3M4M3S2 • Scheda TSX CTC 09

Sezione 3.3
Cablaggio

Cablaggio

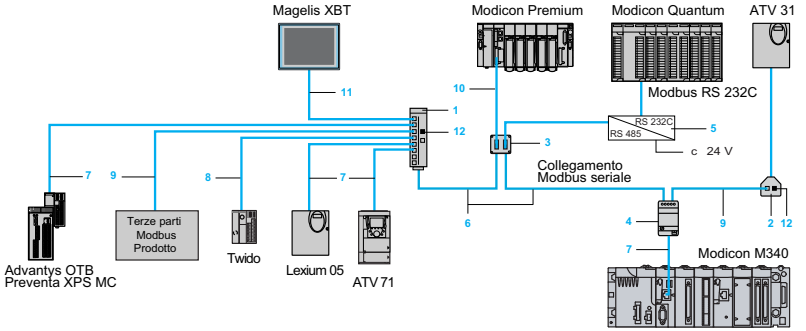
Generale

Per impostare un collegamento seriale sui seguenti processori e sul modulo indicato sono necessari diversi cavi e accessori:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020 e
- modulo BMX NOM 0200.

Sistema di cablaggio

La figura seguente mostra un esempio di collegamento seriale Modicon Modbus M340 e di sistema di cablaggio in modalità carattere. I **cavi** *(vedi pagina 61)* e **gli accessori di collegamento** *(vedi pagina 62)* a cui si fa riferimento nella figura sono descritti nelle seguenti tabelle:



Cavi

Nella seguente tabella sono riportati i cavi compatibili con la comunicazione seriale del modulo e dei processori indicati:

Riferimento figura	Designazione	Lunghezza	Caratteristiche	Codice prodotto
6	Cavo di distribuzione a doppia schermatura con doppini intrecciati RS485	100 m	Due estremità scoperte	TSX CSA 100
6	Cavo di distribuzione a doppia schermatura con doppini intrecciati RS485	200 m	Due estremità scoperte	TSX CSA 200
6	Cavo di distribuzione a doppia schermatura con doppini intrecciati RS485	500 m	Due estremità scoperte	TSX CSA 500
7	Cavo Modbus RS485	0,3 m	Due connettori maschio RJ45	VW3 A8 306 R03
7	Cavo Modbus RS485	1 m	Due connettori maschio RJ45	VW3 A8 306 R10
7	Cavo Modbus RS485	3 m	Due connettori maschio RJ45	VW3 A8 306 R30
-	Cavo Modbus RS485	3 m	• Un connettore maschio RJ45 • Un connettore maschio SUB-D a 15 pin	VW3 A8 306
4	Cavo Modbus RS485	0,3 m	• Un connettore maschio RJ45 • Un connettore mini DIN	TWD XCA RJ003
4	Cavo Modbus RS485	1 m	• Un connettore maschio RJ45 • Un connettore mini DIN	TWD XCA RJ010
4	Cavo Modbus RS485	3 m	• Un connettore maschio RJ45 • Un connettore mini DIN	TWD XCA RJ030
5	Cavo Modbus RS485	3 m	• Un connettore maschio RJ45 • Un'estremità scoperta	VW3 A8 306 D30
9	Cavo Modbus RS485	3 m	• Un connettore in miniatura • Un connettore SUB-D a 15 pin	TSX SCP CM 4630
11	Cavo RS485 per display Magelis XBT e terminale	2,5 m	• Un connettore maschio RJ45 • Un connettore femmina SUB-D a 25 pin **Nota:** Questo cavo non è compatibile con il modulo BMX NOM 0200	XBT-Z938
-	Cavo RS485 per le periferiche alimentate tramite collegamento seriale	3 m	Due connettori maschio RJ45 **Nota:** Questo cavo non è compatibile con il modulo BMX NOM 0200.	XBT-Z9980
-	Cavo RS232 a quattro fili per apparecchiatura terminale dati (DTE)	3 m	• Un connettore maschio RJ45 • Un connettore femmina SUB-D a 9 pin	TCS MCN 3M4F3C2

Riferimento figura	Designazione	Lunghezza	Caratteristiche	Codice prodotto
-	Cavo RS232 a quattro fili per apparecchiatura di terminazione circuito dati (DCE)	3 m	● Un connettore maschio RJ45 ● Un connettore maschio SUB-D a 9 pin	TCS MCN 3M4M3S2
-	Cavo RS232 a quattro fili per apparecchiatura di terminazione circuito dati (DCE)	3 m	● Un connettore maschio RJ45 ● Un connettore maschio SUB-D a 9 pin	TCS XCN 3M4F3S4

Accessori di collegamento

Nella seguente tabella sono riportati gli accessori per il cablaggio compatibili con la comunicazione seriale del modulo e dei processori indicati:

Riferimento figura	Designazione	Caratteristiche	Codice prodotto
1	Scatola splitter Modbus	● Dieci connettori RJ45 ● Una morsettiera a vite	LU9 GC3
2	Scatola di giunzione a T	● Due connettori RJ45 ● Cavo integrato 0,3 m con connettore RJ45 all'estremità	VW3 A8 306 TF03
2	Scatola di giunzione a T	● Due connettori RJ45 ● Cavo integrato 1 m con connettore RJ45 all'estremità	VW3 A8 306 TF10
-	Scatola di giunzione a T passiva	● Tre morsettiere a vite ● Adattatore estremità linea RC	TSX SCA 50
3	Subscriber socket passiva a 2 canali	● Due connettori femmina SUB-D a 15 pin ● Due morsettiere a vite ● Adattatore estremità linea RC	TSX SCA 62
4	Scatola di giunzione a T isolata RS485	● Un connettore RJ45 ● Una morsettiera a vite	TWD XCA ISO
-	Scatola di giunzione a T	Tre connettori RJ45	TWD XCA T3RJ
-	Adattatore Bluetooth / Modbus	● Un adattatore Bluetooth con connettore RJ45 ● Un set cavi per PowerSuite con due connettori RJ45 ● Un set cavi per TwidoSuite con un connettore RJ45 e un connettore mini-DIN ● Un adattatore maschio a 9 pin RJ45/SUB-D per variatori ATV	TWD XCA T3RJ

Riferimento figura	Designazione	Caratteristiche	Codice prodotto
5	Adattatore di linea RS232C/RS485 senza segnali modem	19,2kbit/s	XGS Z24
12	Terminatore di linea per connettore RJ45	• Resistenza da 120 Ω • Capacità da 1 nF	VW3 A8 306 RC
-	Terminatore di linea per morsettiera a vite	• Resistenza da 120 Ω • Capacità da 1 nF	VW3 A8 306 DRC
-	Scheda per dispositivi non standard	• Due connettori maschio SUB-D a 25 pin	XBT ZG999
-	Scheda per dispositivi non standard	• Un connettore maschio SUB-D a 25 pin • Un connettore maschio SUB-D a 9 pin	XBT ZG909
-	Scheda per apparecchiatura terminale dati	• Un connettore maschio SUB-D a 9 pin • Un connettore femmina SUB-D a 25 pin	TSX CTC 07
-	Scheda per apparecchiatura terminale dati	• Un connettore maschio SUB-D a 9 pin • Un connettore maschio SUB-D a 25 pin	TSX CTC 10
-	Scheda per apparecchiatura di terminazione circuito dati (DCE)	• Un connettore femmina SUB-D a 9 pin • Un connettore maschio SUB-D a 25 pin	TSX CTC 09

NOTA: questo elenco di cavi e accessori non è esaustivo.

Parte III

Implementazione software del Modbus Seriale e delle comunicazioni in modalità carattere

In questa sezione

Questa sezione fornisce un'introduzione all'implementazione software delle comunicazioni seriali Modbus e in Modalità carattere tramite il programma Unity Pro.

Contenuto di questa parte

Questa parte contiene i seguenti capitoli:

Capitolo	Titolo del capitolo	Pagina
4	Metodologia di installazione	67
5	Comunicazione Modbus Seriale per i processori BMX P34 1000/2000/2010/20102/2020	69
6	Comunicazione modalità carattere per i processori BMX P34 1000/2000/2010/20102/2020	101
7	Comunicazione Modbus Seriale per BMX NOM 0200	125
8	Comunicazione Modalità carattere per BMX NOM 0200	169
9	Diagnostica Modulo BMX NOM 0200	203
10	Oggetti linguaggio di comunicazioni Modbus e modalità caratteri	209
11	Cambio dinamico di protocollo	243

Capitolo 4
Metodologia di installazione

Introduzione alla fase di installazione

Introduzione

L'installazione del software dei moduli specifici dell'applicazione viene eseguita dai vari editor di Unity Pro:

- nella modalità offline
- in modalità online

Se non si dispone di un processore a cui collegarsi, Unity Pro consente di eseguire un test iniziale utilizzando un simulatore. In questo caso, l'installazione è diversa.

Fasi di installazione in caso di utilizzo di un processore

La tabella seguente illustra le varie fasi di installazione con un processore:

Fase	Descrizione	Modalità
Configurazione del processore	Dichiarazione processore	Offline
	Configurazione della porta seriale del processore	
Configurazione del modulo (se applicabile)	Dichiarazione del modulo	Offline
	Configurazione dei canali del modulo	
	Immissione dei parametri di configurazione	
Dichiarazione delle variabili	Dichiarazione delle variabili di tipo IODDTspecifiche al processore e alle variabili del progetto	Offline (1)
Associazione	Associazione delle variabili degli IODDT ai canali configurati (editor delle variabili)	Offline (1)
Programmazione	Programmazione del progetto	Offline (1)
Generazione	Generazione progetto (analisi e modifica dei collegamenti)	Offline
Trasferimento	Trasferimento di un progetto al PLC	Online
Debug	Debug del progetto dalle schermate di debug e dalle tabelle di animazione	Online
Documentazione	Creazione del file della documentazione e stampa delle varie informazioni correlate al progetto	Online
Funzionamento	Visualizzazione di informazioni varie necessarie alla supervisione del progetto	Online
Legenda:		
(1) Queste fasi possono inoltre essere eseguite online.		

Fasi di installazione con un simulatore

La tabella seguente illustra le varie fasi di installazione con un simulatore:

Fase	Descrizione	Modalità
Configurazione del processore	Dichiarazione processore	Offline
	Configurazione della porta seriale del processore	
Configurazione del modulo (se applicabile)	Dichiarazione del modulo	Offline
	Configurazione dei canali del modulo	
	Immissione dei parametri di configurazione	
Dichiarazione delle variabili	Dichiarazione delle variabili di tipo IODDT specifiche al processore e alle variabili del progetto	Offline (1)
Associazione	Associazione delle variabili degli IODDT ai canali configurati (editor delle variabili)	Offline (1)
Programmazione	Programmazione del progetto	Offline (1)
Generazione	Generazione progetto (analisi e modifica dei collegamenti)	Offline
Trasferimento	Trasferimento di un progetto al simulatore	Online
Simulazione	Simulazione del programma senza ingressi/uscite	Online
Regolazione/debug	Debug del progetto dalle tabelle di animazione	Online
	Modifica del programma e dei parametri di regolazione	
Legenda:		
(1) Queste fasi possono inoltre essere eseguite online.		

Configurazione del processore e del modulo

È possibile accedere ai parametri di configurazione solo dal software Unity Pro.

Creazione della documentazione tecnica

Unity Pro consente di creare una documentazione tecnica **di progetto** *(vedi Unity Pro, Modalità operative)*.

Il formato generico della stampa consiste in:
- Un titolo: numero della parte del modulo e sua posizione,
- Una sezione con l'identificazione del modulo,
- Una sezione per canale con tutti i parametri di un canale.

La stampa è consistente con la configurazione: le informazioni non significative disattivate non vengono stampate.

Capitolo 5
Comunicazione Modbus Seriale per i processori BMX P34 1000/2000/2010/20102/2020

Oggetto del capitolo

Questo capitolo presenta il processo di implementazione software per la comunicazione Modbus Seriale per i processori BMX P34 1000/2000/2010/20102/2020.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
5.1	Generalità	70
5.2	Configurazione della comunicazione Modbus Seriale	77
5.3	Programmazione della comunicazione Modbus Seriale	88
5.4	Debug della comunicazione Modbus Seriale	98

Sezione 5.1
Generalità

Oggetto della sezione

Questa sezione presenta i punti generali relativi alla comunicazione Modbus Seriale e ai relativi servizi.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Informazioni sul Modbus Seriale	71
Prestazioni	72
Accesso ai parametri del collegamento seriale	74

Informazioni sul Modbus Seriale

Introduzione

La comunicazione tramite Modbus permette lo scambio di dati tra tutte le periferiche connesse al bus. Il Modbus Seriale è un protocollo che crea un struttura gerarchica (un master e diversi slave).

Il master gestisce tutti gli scambi in due modi:

- il master scambia con lo slave e attende la risposta;
- il master scambia con tutti gli slave senza attendere risposta (trasmissione generale).

NOTA: Fare attenzione che i due master (sullo stesso bus) non inviino richieste simultaneamente, altrimenti le richieste vanno perdute e ogni report avrà un risultato errato che potrebbe essere 16#0100 (impossibile elaborare la richiesta) o 16#ODFF (slave non presente).

⚠ AVVERTENZA
PERDITA CRITICA DEI DATI Le porte di comunicazione devono essere usate solo per il trasferimento di dati non critici. **Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.**

Prestazioni

In breve

È possibile utilizzare le tabelle seguenti per valutare i tempi di scambio tipici della comunicazione Modbus in base a criteri diversi.

I risultati visualizzati corrispondono al periodo operativo medio della funzione `READ_VAR` in millisecondi.

Definizione di tempo di scambio

Il Tempo di scambio è il tempo che intercorre tra l'attivazione e la fine dello scambio. Questo valore include il tempo di comunicazione del collegamento seriale.

Lo scambio viene creato quando si crea la chiamata della funzione di comunicazione.

Lo scambio termina quando si verifica uno dei seguenti eventi:
- i dati vengono ricevuti
- si verifica un'anomalia
- il timeout scade

TempO di scambio per una parola

La tabella seguente mostra i tempi di scambio per una parola di comunicazione Modbus su un processore BMX P34 2020:

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Durata dello scambio in ms Lo slave Modbus è un BMX P34 1000 ciclico
4800	Ciclica	68
4800	10	72
4800	50	100
9600	Ciclica	35
9600	10	40
9600	50	50
19200	Ciclica	20
19200	10	27
19200	50	50
38400	Ciclica	13
38400	10	20
38400	50	50

Il tempo di scambio è simile per i processori BMX P34 2020 e BMX P34 2000/2010/20102, mentre per il BMX P34 1000 è inferiore del 10%.

Tempo di scambio per 100 parole

La tabella seguente mostra i tempi di scambio per 100 parole di comunicazione Modbus su processore BMX P34 2020:

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Durata dello scambio in ms Lo slave Modbus è un BMX P34 1000 ciclico
4800	Ciclica	500
4800	10	540
4800	50	595
9600	Ciclica	280
9600	10	288
9600	50	300
19200	Ciclica	142
19200	10	149
19200	50	150
38400	Ciclica	76
38400	10	80
38400	50	100

Il tempo di scambio è simile per i processori BMX P34 2020 e BMX P34 2000/2010/20102, mentre per il BMX P34 1000 è inferiore del 10%.

Precisione della misura

Tutti i tempi di scambio indicati in precedenza derivano da misurazioni eseguite con un margine di precisione di +/-10 ms.

Accesso ai parametri del collegamento seriale

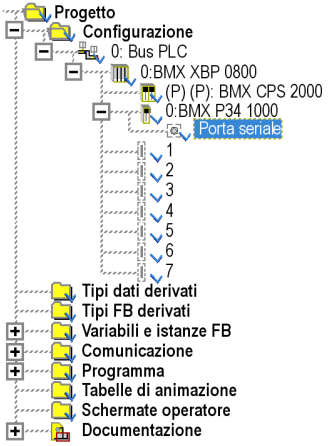
In breve

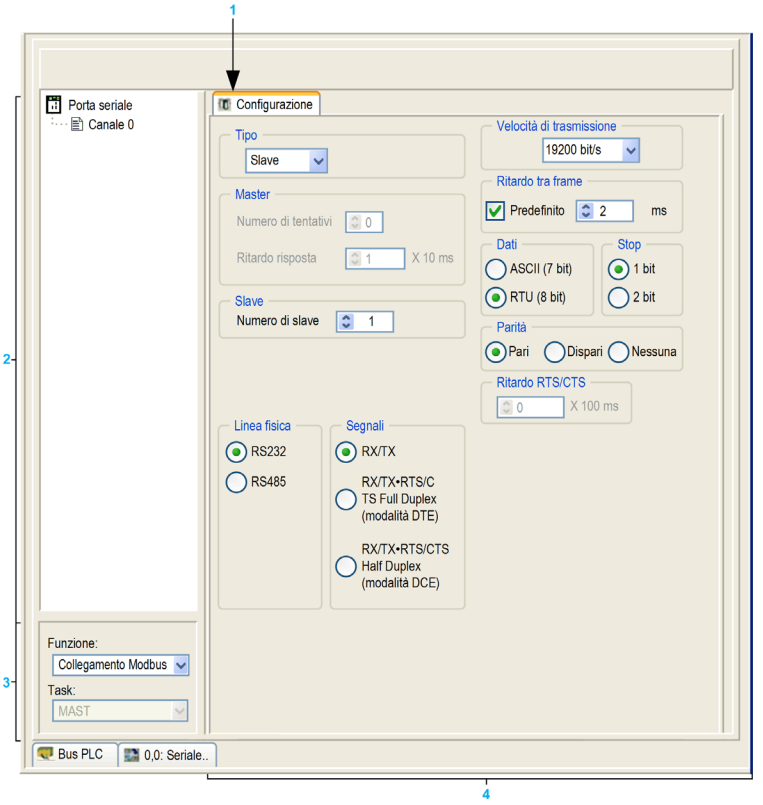
Le sezioni che seguono descrivono l'accesso alla schermata di configurazione della porta seriale e gli elementi generali delle schermate di configurazione e di debug del collegamento Modbus e Modalità caratteri:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020.

Accesso al collegamento seriale

Nella seguente tabella è descritta la procedura per accedere al collegamento seriale:

Passo	Azione
1	Nel browser di progetto, aprire la directory seguente:*Station\Configuration\0: PLC bus\0: rack reference\0: processor reference\SerialPort*. **Risultato**: Viene visualizzata le seguente schermata: 

Passo	Azione
2	Fare doppio clic sulla sottodirectory Porta seriale. **Risultato**: Viene visualizzata la schermata di configurazione: 

Descrizione della schermata di configurazione

La seguente tabella mostra gli elementi della schermata di configurazione:

Indirizzo	Elemento	Funzione
1	Schede	La scheda in primo piano indica la modalità corrente. Ciascuna modalità può essere selezionata tramite la scheda corrispondente. Le modalità disponibili sono: ● Configurazione ● Schermata Debug (accessibile solo in modalità online)

2	Zona canale	Consente di:
		● Scegliere tra la porta seriale e il canale 0 facendo clic su una o sull'altro
		● Visualizzare le schede seguenti facendo clic sulla porta seriale:
		  ● "Descrizione", che mostra le caratteristiche del dispositivo
		  ● **"Oggetti di I/O"** *(vedi Unity Pro, Modalità operative)*, utilizzato per presimbolizzare gli oggetti di ingresso/uscita
		● Visualizzare le schede seguenti facendo clic sul canale:
		  ● Configurazione
		  ● Debug
		● Visualizzare il simbolo e il nome del canale definito dall'utente utilizzando l'editor delle variabili.

3	Zona parametri generali	Consente di selezionare i parametri generali associati al canale: ● Funzione: Le funzioni disponibili sono Modbus e modalità carattere: la configurazione predefinita è con la funzione Modbus. ● Task: Definisce il master task in cui verranno scambiati gli oggetti di scambio impliciti del canale. Quest'area è disabilitata e non è configurabile.
4	Zona di configurazione o debug	In modalità configurazione, viene usata per configurare i parametri del canale. In modalità debug, viene usata per eseguire il debug del canale di comunicazione.

Sezione 5.2
Configurazione della comunicazione Modbus Seriale

Oggetto della sezione

Questa sezione descrive il processo di configurazione software per la comunicazione Modbus Seriale.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Schermata di configurazione della comunicazione Modbus Seriale	78
Funzioni accesibili Modbus	80
Valori predefiniti dei parametri di comunicazione Modbus Seriale	81
Parametri Modbus legati all'applicazione	82
Parametri Modbus legati alla trasmissione	84
Parametri del segnale e della linea fisica in Modbus	86

Schermata di configurazione della comunicazione Modbus Seriale

Generale

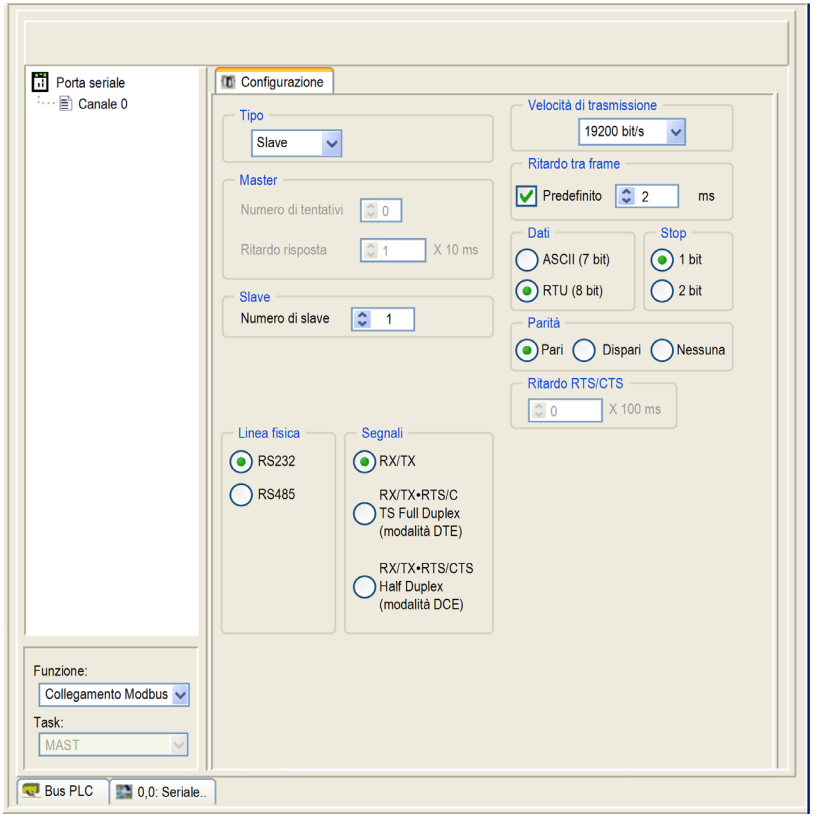
Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modbus seriale.

Accesso alla schermata di configurazione

Per accedere alla schermata di configurazione di comunicazione Modbus seriale, aprire la directory Porta seriale nel browser del progetto *(vedi pagina 74)*.

Schermata di configurazione della comunicazione Modbus Seriale

La figura mostra la schermata di configurazione predefinita della comunicazione Modbus seriale:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione del collegamento Modbus:

Legenda	Elemento	Commento
1	Parametri dell'applicazione *(vedi pagina 82)*	Tali parametri sono accessibili da tre zone: ● Tipo, ● Master, ● Slave.
2	Parametri di trasmissione *(vedi pagina 84)*	Tali parametri sono accessibili da cinque zone: ● Velocità di trasmissione, ● Ritardo tra frame, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 86)*	Tali parametri sono accessibili da tre zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS.

NOTA: Durante la configurazione della comunicazione Modbus Seriale in modalità Master, la zona Slave è disattivata e non può essere modificata e viceversa.

Funzioni accesibili Modbus

In breve

L'accessibilità delle funzioni per la configurazione del link seriale dei seguenti processori tramite Modbus seriale dipende dal collegamento fisico utilizzato:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020.

Funzioni accessibili

La tabella seguente mostra le varie funzioni configurabili in base al tipo di collegamento seriale utilizzato.

Funzione	Collegamento RS 485	Collegamento RS 232
Numero tentativi master	X	X
Periodo risposta master	X	X
Numero di slave	X	X
Velocità di trasmissione	X	X
Ritardo tra frame	X	X
Dati	• ASCII (7 bit) • RTU (8 bit)	• ASCII (7 bit) • RTU (8 bit)
Stop	• 1 bit • 2 bit	• 1 bit • 2 bit
Parità	• Odd • Pari • Nessuno	• Odd • Pari • Nessuno
Segnali RX/TX	X	X
Segnali RTS/CTS	-	X
Ritardo RTS/CTS	-	X

X Funzione accessibile
- Funzione inaccessibile

Valori predefiniti dei parametri di comunicazione Modbus Seriale

In breve

Tutti i parametri di comunicazione Modbus Seriale presentano valori predefiniti.

Valori predefiniti

La tabella seguente mostra i valori predefiniti per i parametri di comunicazione Modbus Seriale:

Parametro di configurazione	esad.
Modalità	Slave
Linea fisica	RS232
Numero di slave	1
Ritardo tra frame	2 ms
Velocità di trasmissione	19200 bit/s
Parità	Pari
Bit di dati	RTU (8 bit)
Bit di stop	1 bit

Parametri Modbus legati all'applicazione

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri dell'applicazione.

Tali parametri sono accessibili da tre zone di configurazione:
- La zona Tipo,
- La zona Master,
- La zona Slave.

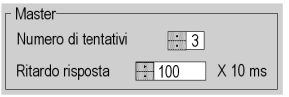
La zona Tipo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di selezionare il tipo di Modbus Seriale da utilizzare.
- **Master**: Quando la stazione coinvolta è il master.
- **Slave**: Quando la stazione coinvolta è lo slave.

La zona Master

La zona di configurazione illustrata di seguito è accessibili solo se si seleziona "Master" nella zona "Tipo":



Questa zona consente di specificare i parametri seguenti:
- **Numero di tentativi**: numero di tentativi di connessione eseguiti dal master prima di considerare lo slave assente.
  il valore predefinito è 3;
  intervallo di valori possibili da 1 a 15;
  un valore di 0 indica nessun tentativo da parte del master.
- **Ritardo risposta**: il tempo tra la richiesta iniziale del master e un tentativo ripetuto in assenza di risposta dello slave. È il tempo massimo tra la trasmissione dell'ultimo carattere della richiesta del master e la ricezione del primo carattere della richiesta inviato dallo slave.
  Il valore predefinito è 1 secondo (100*10ms).
  intervallo di valori possibili da 10 ms a 10 s.

NOTA: Il ritardo della Risposta del Master deve essere almeno uguale al tempo di ritardo alla Risposta più lungo degli Slaves presenti sul bus.

La zona Slave

La zona di configurazione illustrata di seguito è accessibili solo se si seleziona "Slave" nella zona "Tipo":

Questa zona consente di specificare il numero di slave del processore.

il valore predefinito è 1;

intervallo di valori possibili da 1 a 247.

NOTA: in una configurazione Modbus Slave, è possibile utilizzare un indirizzo aggiuntivo, il numero 248, per la comunicazione seriale da punto a punto.

Parametri Modbus legati alla trasmissione

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di trasmissione.

Tali parametri sono accessibili da cinque zone:
- La zona Velocità di trasmissione
- La zona Ritardo tra caratteri;
- La zona Dati,
- La zona Stop,
- La zona Parità.

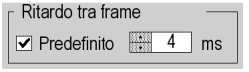
La zona Velocità di trasmissione

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

È possibile utilizzarla per selezionare la velocità di trasmissione del Modbus Seriale. La velocità selezionata deve essere coerente con gli altri dispositivi. I valori configurabili sono 300, 600, 1200, 2400, 4800, 9600, 19200 e 38400 bit al secondo.

L'area Ritardo tra frame

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



Il Ritardo tra frame è l'intervallo minimo di separazione di ricezione di due frame. Tale ritardo viene gestito quando il PLC (master o slave) sta ricevendo dei messaggi.

NOTA: il valore predefinito dipende dalla velocità di trasmissione selezionata.

NOTA: Il ritardo tra le frame dovrebbe essere rappresentato dal valore Predefinito in modo da essere conforme al Modbus. Nel caso in cui uno Slave non sia conforme, il valore può essere cambiato e deve essere identico sia per Master che per tutti gli Slave del Bus.

La zona Dati

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di specificare il tipo di codifica utilizzato per comunicare con il Modbus Seriale. Questo campo è impostato in base agli altri dispositivi collegati sul bus. Sono possibili due modalità di configurazione:

- Modalità RTU:
  - i caratteri sono codificati su 8 bit;
  - la fine del frame viene individuata in presenza di un silenzio di almeno 3,5 caratteri;
  - L'integrità del frame viene controllata tramite una parola nota come checksum CRC , che è contenuta nel frame.
- modalità ASCII:
  - i caratteri sono codificati su 7 bit;
  - l'inizio del frame viene individuato alla ricezione del carattere **":"** ;
  - la fine del frame è individuata da un ritorno a capo e da un avanzamento riga;
  - L'integrità del frame viene controllata tramite un byte noto come checksum LRC , che è contenuto nel frame.

La zona Stop

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

La zona Stop consente di inserire il numero di bit di stop utilizzati per la comunicazione. Questo campo viene definito in base agli altri dispositivi. I valori configurabili sono:

- 1 bit
- 2 bit

La zona Parità

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Queste zone consentono di stabilire l'aggiunta o meno di un bit di parità e di definirne il tipo. Questo campo viene definito in base agli altri dispositivi. I valori configurabili sono:

- Pari
- Odd
- Nessuno

Parametri del segnale e della linea fisica in Modbus

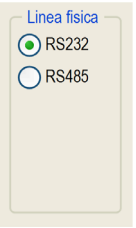
In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di segnale e linea fisica.

Tali parametri sono accessibili da tre zone:
- La zona Linea fisica,
- La zona Segnali,
- la zona Ritardo RTS/CTS.

La zona Linea fisica

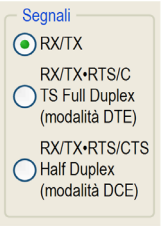
Questa zona di configurazione appare sullo schermo come illustrato di seguito:



In questa zona, è possibile scegliere due tipi di linea fisica per la porta seriale sui processori BMX P34 1000/2000/2010/20102/2020:
- La linea RS 232,
- La linea RS 485.

La zona Segnali

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



In questa zona, è possibile selezionare i segnali supportati dalla linea fisica RS 232:

- RX/TX
- RX/TX + RTS/CTS DTEmode
- RX/TX + RTS/CTS DCEmode

Se è configurata la linea RS 485, tutta la zona sarà disattivata e il valore predefinito sarà RX/TX.

NOTA: sono disponibili solo i segnali RX/TX e RX/TX + RTS/CTS quando si configura la porta seriale per i processori BMX P34 1000/2000/2010/20102/2020.

La RTS/CTS zona Ritardo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

LaRTS/CTS zona ritardo è disponibile solamente quando entrambe le caselle di selezione RS232 e RX/TX+RTS/CTS sono selezionate. Viene selezionato un algoritmo di controllo di flusso RTS/CTS se il valore predefinito è 0 ms. Un valore diverso da 0 consente un algoritmo di controllo del modem RTS/CTS .

L'algoritmo di RTS/CTS controllo del flusso (DTE <-> DTE) è diverso dall'algoritmo RTS/CTS di controllo del modem (DTE <-> DCE) come segue:

- L'algoritmo di controllo del flusso RTS/CTS è relativo al buffer di ricezione dell'overflow (full duplex).
- L'algoritmo di controllo del modem RTS/CTS si occupa del processo di trasmissione condiviso, es. un modem radio.

RTS/CTS Algoritmo del controllo di flusso

Lo scopo è di prevenire un overflow del buffer di ricezione.

Il segnale in uscita RTS di ciascuna periferica è collegato al segnale in ingresso CTS dell'altra periferica. Il trasmettitore (M340) è autorizzato a trasmettere i dati durante la ricezione del segnale in ingresso RTS (es. un altro M340) sul suo ingresso CTS . Questo algoritmo è simmetrico e consente la comunicazione asincrona full duplex.

RTS/CTS Algoritmo del controllo del modem

Prima che venga trasmessa una richiesta, il mittente (M340) attiva il segnale RTS e aspetta che il segnale CTS venga attivato dal modem. Se il CTS non viene attivato dopo il ritardo RTS/CTS , la richiesta viene abbandonata.

Sezione 5.3
Programmazione della comunicazione Modbus Seriale

Oggetto della sezione

Questa sezione descrive il processo di programmazione utilizzato per implementare la comunicazione Modbus seriale.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Servizi supportati da un processore master per collegamento Modbus	89
Servizi supportati da un processore slave del collegamento Modbus	96

Servizi supportati da un processore master per collegamento Modbus

In breve

Se utilizzati come processori master in un collegamento Modbus, i seguenti processori supportano diversi servizi tramite le funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH.

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020.

Scambi di dati

La lettura o scrittura delle variabili viene effettuata indirizzando le seguenti richieste al dispositivo slave di destinazione.

Tali richieste utilizzano le funzioni di comunicazione READ_VAR ,WRITE_VAR e DATA_EXCH:

Richiesta Modbus	Codice funzione	Funzione di comunicazione
Lettura bit	16#01 o 16#02	READ_VAR
Lettura parole	16#03 o 16#04	READ_VAR
Scrittura bit	16#0F	WRITE_VAR
Scrittura parole	16#10	WRITE_VAR
Altra richiesta	tutto	DATA_EXCH

NOTA: WRITE_VAR può essere usato nella modalità broadcast, mentre READ_VAR non può essere usato in questa modalità. In questo caso, il PLC non riceve alcuna risposta. L'invio di una richiesta broadcast implica la reimpostazione del bit di attività e la restituzione del codice 16#01 (Arresto dello scambio su timeout) nella seconda parola di gestione EF.

NOTA: Gli oggetti letti dal PLC Modicon M340 possono essere di tipo %I e %IW. In questo caso, la funzione READ_VAR genera una richiesta Modbus: FC 0x2 o 0x4. In un PLC Quantum, permette di accedere agli Stati d'ingresso o ai registri di Stato d'ingresso.

Più in generale, è possibile inviare qualsiasi richiesta Modbus a un dispositivo slave tramite la funzione di comunicazione DATA_EXCH .

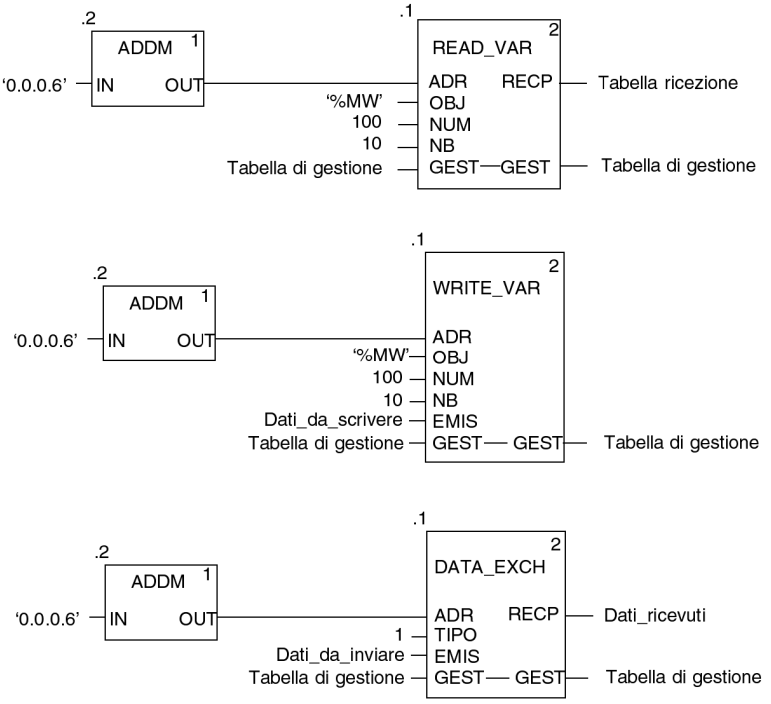
Funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH

Per inviare e ricevere i dati tramite un canale di comunicazione Modbus sono definite tre funzioni di comunicazione specifiche:

- READ_VAR: Per leggere le variabili
- WRITE_VAR: Per scrivere le variabili
- DATA_EXCH: Per inviare le richieste Modbus a un altro dispositivo sul protocollo selezionato

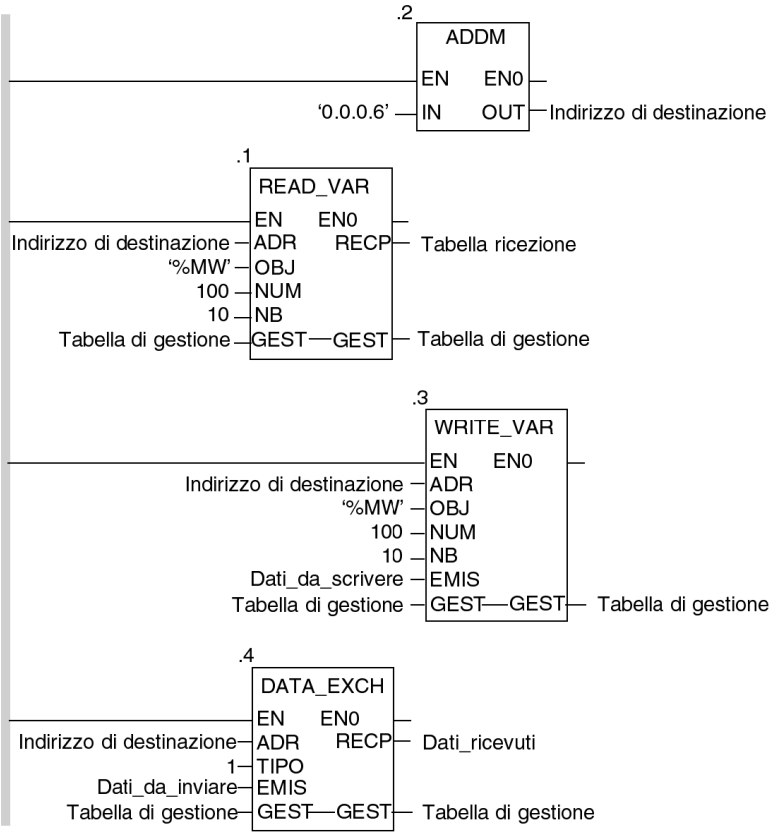
Esempio Programmazione in FBD

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio FBD :



Esempio Programmazione in Ladder

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio Ladder:



Esempio di programmazione in ST

Le righe di codice di seguito rappresentano un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio ST :

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,
Receiving_Table);

WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,
Management_Table);

DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,
Received_data);
```

Annullamento di uno scambio

Uno scambio eseguito dalle funzioni READ_VAR, WRITE_VAR e DATA_EXCH può essere annullato secondo due modi di programmazione, presentati in linguaggio ST di seguito:

- Uso della funzione CANCEL:

```
IF (%MW40.0) THEN
 %MW200:=SHR(%MW40,8;)
 CANCEL(%MW200,%MW185);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione READ_VAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma esegue le istruzioni seguenti:
  - Sposta i bit %MW40 di un byte (8 bit) a destra e carica il byte corrispondente al numero di scambio della comunicazione nella parola %MW200 ,
  - annulla lo scambio il cui numero è contenuto nella parola %MW200 mediante la funzione CANCEL.

- Tramite il bit di annullamento della funzione di comunicazione:

```
IF (%MW40.0) THEN
 SET(%MW40.1);
 READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4, %MW10:10);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione READ_VAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma imposta il bit %MW40.1, il bit di annullamento della funzione, a 1. Quest'azione interrompe la comunicazione della funzione READ_VAR.

NOTA: Quando si utilizza il bit di annullamento della funzione di comunicazione contenuto nella parola di gestione scambio (%MW40, in questo esempio), per poter attivare l'annullamento dello scambio deve essere chiamata la funzione (READ_VAR, in questo esempio).

NOTA: quando si utilizza il bit di annullamento della funzione di comunicazione, è possibile annullare una comunicazione da una tabella di animazione Questo può essere fatto semplicemente impostando il bit di annullamento funzione a 1 (%MW40.1, in questo esempio) e riavviando di nuovo la funzione di comunicazione.

NOTA: Questo esempio di programmazione è relativo alla funzione READ_VAR , ma è applicabile anche alla funzione WRITE_VAR e alle funzioni DATA_EXCH .

NOTA: la funzione CANCEL utilizza una parola di rapporto (%MW185 in questo esempio).

Descrizione dei parametri della funzione ADDM

La tabella seguente descrive i diversi parametri della funzione ADDM:

Parametro	Tipo	Descrizione
IN	STRING	Indirizzo del dispositivo su bus o collegamento seriale. La sintassi dell'indirizzo è del tipo 'r.m.c.node'. L'indirizzo è composto dai seguenti parametri: ● **r**: Numero di rack del processore, sempre =0, ● **m**: Numero di slot del processore all'interno del rack, sempre=0 ● **c**: Numero di canale, sempre=0 in quanto il collegamento seriale di un processore è sempre canale 0 ● **node**: Numero di slave a cui è stata inviata la richiesta.
OUT	ARRAY [0..7] OF INT	Array che rappresenta l'indirizzo di un dispositivo. Questo parametro può essere usato come un parametro d'ingresso per diverse funzioni di comunicazione.

Descrizione dei parametri della funzione READ_VAR

La tabella seguente descrive i diversi parametri della funzione READ_VAR:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
OBJ	STRING	Tipo di oggetto da leggere: I tipi disponibili sono indicate di seguito: ● %M: bit interno ● %MW: parola interna ● %I: bit di ingresso esterno ● %IW: parola di ingresso esterna
NUM	DINT	Indirizzo del primo oggetto da leggere:
NB	INT	Numero di oggetti consecutivi da leggere.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio,
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

RECP	ARRAY [n..m] OF INT	Tabella di parole contente il valore degli oggetti letti.

Descrizione dei parametri della funzione WRITE_VAR

La tabella seguente descrive i diversi parametri della funzione WRITE_VAR:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
OBJ	STRING	Tipo dell'oggetto da scrivere: I tipi disponibili sono indicate di seguito: ● %M: bit interno ● %MW: parola interna **Nota:** WRITE_VAR non può essere utilizzato per le variabili %I e %IW.
NUM	DINT	Indirizzo del primo oggetto da scrivere.
NB	INT	Numero di oggetti consecutivi da scrivere:
EMIS	ARRAY [n..m] OF INT	Tabella di parole contente il valore degli oggetti da scrivere.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio,
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

Descrizione dei parametri della funzione DATA_EXCH

La tabella seguente descrive i diversi parametri della funzione DATA_EXCH:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
TIPO	INT	Per i PLC Modicon M340, l'unico valore possibile è: **1**: Trasmissione di un array **EMIS** , quindi il PLC attende la ricezione di un array **RECP** .
EMIS	ARRAY [n..m] OF INT	Tabella degli integrali da inviare al dispositivo di destinazione della richiesta. **Nota:** È imperativo che la lunghezza dei dati da inviare (byte) sia assegnata alla quarta parola della tabella di gestione prima di lanciare la funzione, affinché questa sia eseguita correttamente.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio,
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

RECP	ARRAY [n..m] OF INT	Tabella degli integrali contenente i dati ricevuti. **Nota:** La dimensione dei dati ricevuti (in byte) viene scritta automaticamente dal sistema nella quarta parola della tabella di gestione.

Servizi supportati da un processore slave del collegamento Modbus

In breve

Se vengono utilizzati come processore slave in un collegamento Modbus, i seguenti processori supportano diversi servizi:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020.

Scambi di dati

Un processore slave gestisce le richieste seguenti:

Richiesta Modbus	Codice funzione	Oggetto PLC
Lettura di n bit di uscita	16#01	%M
Lettura di n parole di uscita	16#03	%MW
Scrittura di n bit di uscita	16#0F	%M
Scrittura di n parole di uscita	16#10	%MW

Diagnostica e manutenzione

Le informazioni su diagnostica e manutenzione accessibili da un collegamento Modbus sono elencate di seguito:

Designazione	Codice funzione/Codice funzione secondaria
Eco	16#08 / 16#00
Lettura dei registri di diagnostica del PLC	16#08 / 16#02
Reset dei registri di diagnostica del PLC e azzeramento dei contatori	16#08 / 16#0A
Lettura numero di messaggi sul bus	16#08 / 16#0B
Lettura numero di errori di comunicazione rilevati sul bus	16#08 / 16#0C
Lettura numero di errori di eccezione rilevati sul bus	16#08 / 16#0D
Lettura del numero di messaggi ricevuti dallo slave	16#08 / 16#0E
Lettura del numero di mancate risposte dallo slave	16#08 / 16#0F
Lettura numero di riconoscimenti negativi dallo slave.	16#08 / 16#10
Lettura numero di risposte di eccezione dallo slave.	16#08 / 16#11
Lettura numero di overflow carattere sul bus	16#08 / 16#12

Designazione	Codice funzione/Codice funzione secondaria
Lettura contatore di evento	16#0B
Lettura evento di connessione	16#0C
Lettura identificazione	16#11
Lettura di identificazione del dispositivo	16#2B / 16#0E

Sezione 5.4
Debug della comunicazione Modbus Seriale

Schermata di debug della comunicazione seriale Modbus

Generale

La schermata di debug della comunicazione Modbus seriale è accessibile solo in modalità online.

Accesso alla schermata di debug

La tabella seguente descrive la procedura per accedere alla schermata di debug della comunicazione Modbus seriale:

Passo	Azione
1	Accedere alla schermata di configurazione della comunicazione Modbus seriale. *(vedi pagina 78)*
2	Selezionare la scheda "Debug" sulla schermata visualizzata.

Descrizione della schermata Debug

La schermata di debug è suddivisa in due zone:
- La zona Tipo,
- La zona Contatori.

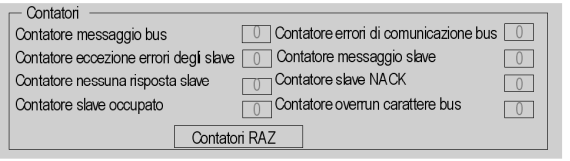
La zona Tipo

La zona è illustrata di seguito:

Indica il tipo di funzione Modbus configurata (in questo caso, Master).

La zona Contatori

La zona è illustrata di seguito:



Questa zona mostra i vari contatori di debug.

Il pulsante Azzera contatori azzera tutti i contatori della modalità di debug.

Funzionamento dei contatori

I contatori di debug della comunicazione Modbus seriale sono i seguenti:

- Contatore messaggio bus: questo contatore indica il numero di messaggi individuati dal processore sul collegamento seriale. I messaggi con un risultato di controllo CRC negativo non vengono contati.
- Contatore errori di comunicazione bus: questo contatore indica il numero di risultati di controllo CRC negativi contati dal processore. Se viene rilevato un errore di carattere (overflow, errore di parità), oppure se il messaggio è più corto di 3 byte, il sistema che riceve i dati non può eseguire il controllo CRC. In tali casi, il contatore viene incrementato di conseguenza.
- Contatore di errore eccezione slave: questo contatore indica il numero di errori di eccezione Modbus individuati dal processore.
- Contatore messaggio slave: questo contatore indica il numero di messaggi ricevuti ed elaborati dal collegamento Modbus.
- Contatore "nessuna risposta" slave: questo contatore indica il numero di messaggi inviati dal sistema remoto per cui non ha ricevuto risposta (normale o di eccezione). Conteggia, inoltre, il numero di messaggi ricevuti in modalità trasmissione.
- Contatore riconoscimento slave negativo: questo contatore indica il numero di messaggi inviati al sistema remoto per cui è stato restituito un riconoscimento negativo.
- Contatore slave occupato: questo contatore indica il numero di messaggi inviati al sistema remoto per cui è stato restituito un messaggio di eccezione "slave occupato".
- Contatore overflow carattere bus: questo contatore indica il numero di messaggi inviati al processore che non è stato possibile acquisire a causa di un overflow di caratteri sul bus. L'overflow è causato da:
  - dati di tipo carattere che vengono trasmessi sulla porta seriale più rapidamente di quanto non sia possibile memorizzarli,
  - Perdita dei dati dovuta a un'anomalia hardware.

NOTA: per tutti i contatori, il conteggio inizia dall'avvio più recente, dall'operazione di azzeramento contatori o dall'accensione del processore.

Capitolo 6
Comunicazione modalità carattere per i processori BMX P34 1000/2000/2010/20102/2020

Oggetto della sezione

Questo capitolo presenta l'implementazione del software di comunicazione con la Modalità Carattere per i processori BMX P34 1000/2000/2010/20102/2020.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
6.1	Generalità	102
6.2	Configurazione di comunicazione in modalità caratteri	106
6.3	Programmazione comunicazione in modalità caratteri	117
6.4	Debugging della comunicazione in modalità caratteri	123

Sezione 6.1
Generalità

Oggetto della sezione

Questa sezione fornisce una panoramica dei punti generali relativi alla comunicazione in modalità carattere e ai relativi servizi.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Informazioni sulla comunicazione in modalità carattere	103
Prestazioni	104

Informazioni sulla comunicazione in modalità carattere

Introduzione

La comunicazione in modalità carattere consente l'esecuzione di funzioni di comunicazione e dialogo tra i PLC e i dispositivi seguenti:

- periferiche normali (stampanti, schermi tattili, terminali di laboratorio, ecc.),
- periferiche specializzate (lettori di codice a barre, ecc.),
- calcolatori (controllo, gestione produzione, ecc.),
- dispositivi diversi (comandi numerici, controllori a velocità variabile, ecc.),
- Modem esterno.

⚠ AVVERTENZA
PERDITA CRITICA DEI DATI Le porte di comunicazione devono essere usate solo per il trasferimento di dati non critici. **Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.**

Prestazioni

In breve

Le seguenti tabelle descrivono i tempi di scambio tipici della Modalità caratteri.

I risultati visualizzati corrispondono al periodo operativo medio della funzione `PRINT_CHAR` in millisecondi.

Definizione di tempo di scambio

Il Tempo di scambio è il tempo che intercorre tra l'attivazione e la fine dello scambio. Questo valore include il tempo di comunicazione del collegamento seriale.

Lo scambio viene creato quando si crea la chiamata della funzione di comunicazione.

Lo scambio termina quando si verifica uno dei seguenti eventi:
- Ricezione dei dati
- Anomalia
- Scadenza timeout

Tempi di scambio per 80 caratteri

Nella seguente tabella sono indicati i dati di scambio per la trasmissione di 80 caratteri in Modalità caratteri su un processore BMX P34 2020:

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Durata dello scambio in ms
1200	10	805
1200	20	820
1200	50	850
1200	100	900
1200	255	980
4800	10	210
4800	20	220
4800	50	250
4800	100	300
4800	255	425
9600	10	110
9600	20	115
9600	50	145
9600	100	200
9600	255	305
19200	10	55

19200	20	60
19200	50	95
19200	100	100
19200	255	250

I tempi di scambio del processore BMX P34 2000/2010/20102 sono simili a quelli del processore BMX P34 2020. I tempi di scambio del BMX P34 1000 sono inferiori del 10%.

Precisione della misura

Tutti i tempi di scambio indicati in precedenza derivano da misurazioni eseguite con un margine di precisione di +/-10 ms.

Sezione 6.2
Configurazione di comunicazione in modalità caratteri

Oggetto della sezione

Questa sezione descrive il processo di configurazione utilizzato per implementare la comunicazione in modalità carattere.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Schermata di configurazione della comunicazione in modalità carattere	107
Funzioni accessibili in modalità carattere	109
Valori predefiniti dei parametri di comunicazione in modalità carattere	110
Parametri di rilevamento fine messaggio in modalità caratteri	111
Parametri di trasmissione in modalità carattere	113
Parametri del segnale e della linea fisica in modalità carattere	115

Schermata di configurazione della comunicazione in modalità carattere

Generale

Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modalità Carattere.

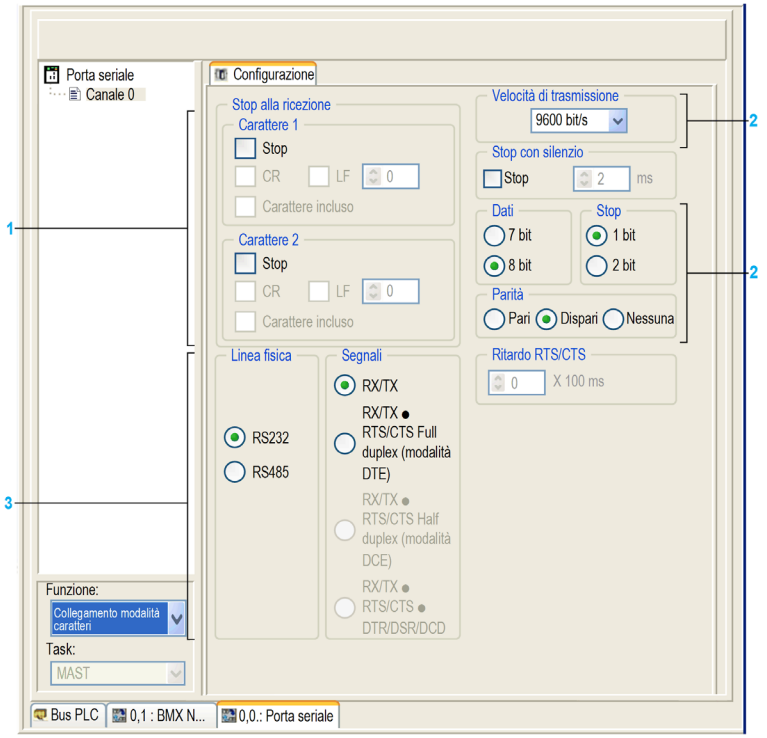
Accesso alla schermata di configurazione

La tabella seguente descrive la procedura per accedere alla schermata di configurazione della comunicazione in modalità carattere.

Passo	Azione
1	Aprire la sottodirectory Porta seriale nel browser progetto *(vedi pagina 74)*.
2	Selezionare la funzione Collegamento modalità carattere sulla schermata visualizzata.

Schermata di configurazione della comunicazione in modalità carattere

La figura mostra la schermata di configurazione predefinita della comunicazione in modalità carattere:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione della Modalità Carattere:

Legenda	Elemento	Commento
1	Parametri di individuazione fine messaggio *(vedi pagina 111)*	Tali parametri sono accessibili da due zone: ● Stop alla ricezione, ● Stop con silenzio.
2	Parametri di trasmissione *(vedi pagina 113)*	Tali parametri sono accessibili da quattro zone: ● Velocità di trasmissione, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 115)*	Tali parametri sono accessibili da tre zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS.

Funzioni accessibili in modalità carattere

In breve

L'accessibilità delle funzioni per la configurazione del link seriale dei seguenti processori tramite il protocollo Modalità caratteri dipende dal collegamento fisico utilizzato:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020.

Funzioni accessibili

La tabella seguente mostra le varie funzioni configurabili in base al tipo di collegamento seriale utilizzato.

Funzione	Collegamento RS 485	Collegamento RS 232
Velocità di trasmissione	X	X
Dati	• 7 bit • 8 bit	• 7 bit • 8 bit
Stop	• 1 bit • 2 bit	• 1 bit • 2 bit
Parità	• Odd • Pari • Nessuno	• Odd • Pari • Nessuno
Stop alla ricezione	X	X
Stop con silenzio	X	X
Segnali RX/TX	X	X
Segnali RTS/CTS	-	X
Ritardo RTS/CTS	-	X

X Funzione accessibile
- Funzione inaccessibile

Valori predefiniti dei parametri di comunicazione in modalità carattere

In breve

Tutti i parametri di comunicazione in modalità carattere presentano valori predefiniti.

Valori predefiniti

La tabella seguente mostra i valori predefiniti per i parametri di comunicazione in modalità carattere:

Parametro di configurazione	Valore
Linea fisica	RS 232
Velocità di trasmissione	9600 bit/s
Parità	Dispari
Bit di dati	8 bit
Bit di stop	1 bit

Parametri di rilevamento fine messaggio in modalità caratteri

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di individuazione fine messaggio.

Tali parametri sono accessibili da due zone:

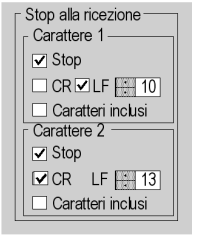
- La zona Stop alla ricezione: si arresta alla ricezione di un carattere speciale;
- La zona Stop con silenzio: si arresta in presenza di silenzio.

Condizioni di utilizzo

Se si seleziona Stop con silenzio, si deseleziona lo Stop alla ricezione e viceversa.

La zona Stop alla ricezione

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



È possibile terminare una richiesta di ricezione dopo la ricezione di un carattere specifico.

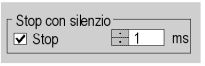
Selezionando l'opzione Stop, è possibile configurare Stop alla ricezione in modo da attivarla da uno specifico carattere di fine messaggio:

- CR: consente di individuare la fine del messaggio tramite un ritorno a capo;
- LF: consente di individuare la fine del messaggio tramite un avanzamento di riga;
- Campo inserimento dati: consente di identificare un carattere di fine messaggio diverso dai caratteri CR o LF, mediante un valore decimale:
  - tra 0 e 255 se i dati sono codificati a 8 bit
  - tra 0 e 127 se i dati sono codificati a 7 bit
- Carattere incluso: consente di includere un carattere di fine messaggio nella tabella di ricezione dell'applicazione del PLC.

È possibile configurare due caratteri di fine ricezione. Nella finestra seguente, il termine della ricezione di un messaggio è individuato da un carattere LF o CR.

La zona Stop con silenzio

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



Questa zona consente di individuare la fine di un messaggio in ricezione tramite l'assenza dei caratteri di fine messaggio in un intervallo determinato.

Stop con silenzio viene convalidata selezionando la casella Stop. La durata del silenzio (espressa in millisecondi) viene impostata tramite il campo di inserimento dati.

NOTA: i valori disponibili sono compresi tra 1ms e 10000ms e dipendono dalla velocità di trasmissione selezionata.

Parametri di trasmissione in modalità carattere

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di trasmissione.

Tali parametri sono accessibili da quattro zone:
- La zona Velocità di trasmissione
- La zona Dati,
- La zona Stop,
- La zona Parità.

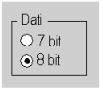
La zona Velocità di trasmissione

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

È possibile utilizzarla per selezionare la velocità di trasmissione del protocollo Modalità carattere. La velocità selezionata deve essere coerente con gli altri dispositivi. I valori configurabili sono 300, 600, 1200; 2400, 4800, 9600, 19200 e 38400 bit al secondo.

La zona Dati

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

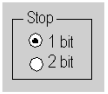


In questa zona è possibile specificare la dimensione dei dati da scambiare sul collegamento. I valori disponibili sono:
- 7 bit
- 8 bit

All'utente viene richiesto di regolare il numero di bit di dati in base al dispositivo remoto utilizzato.

La zona Stop

La zona è illustrata di seguito:



La zona Stop consente di inserire il numero di bit di stop utilizzati per la comunicazione. All'utente viene richiesto di regolare il numero di bit di stop in base al dispositivo remoto utilizzato.

I valori configurabili sono:

- 1 bit
- 2 bit

La zona Parità

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di stabilire l'aggiunta o meno di un bit di parità e di definirne il tipo. All'utente viene richiesto di regolare la parità in base al dispositivo remoto utilizzato. I valori configurabili sono:

- Pari
- Odd
- Nessuno

Parametri del segnale e della linea fisica in modalità carattere

In breve

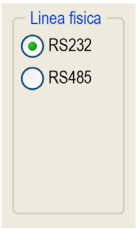
Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di segnale e linea fisica.

Tali parametri sono accessibili da tre zone:

- La zona Linea fisica
- La zona Segnali
- La zona Ritardo RTS/CTS

La zona Linea fisica

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

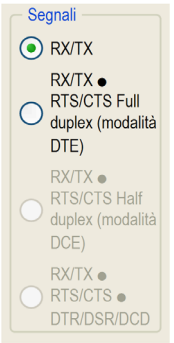


In questa zona, è possibile scegliere due tipi di linea fisica per la porta seriale sui processori BMX P34 1000/2000/2010/20102/2020:

- La linea RS 232
- La linea RS 485

La zona Segnali

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



In questa zona, è possibile selezionare i segnali supportati dalla linea fisica RS 232:

- RX/TX
- RX/TX + RTS/CTS Full Duplex (modalità DTE)

Se è configurata la linea RS 485, tutta la zona sarà disattivata e il valore predefinito sarà RX/TX.

NOTA: Quando si configura la porta seriale per i processori BMX P34 1000/2000/2010/20102/2020, sono disponibili solo i segnali RX/TX e RX/TX + RTS/CTS.

La RTS/CTS zona Ritardo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

LaRTS/CTS zona ritardo è disponibile solamente quando entrambe le caselle di selezione RS232 e RX/TX+RTS/CTS sono selezionate.

Viene selezionato un algoritmo di controllo del flusso RTS/CTS : prima di trasmettere una stringa di caratteri, il sistema attende l'attivazione del segnale CTS (Clear To Send) . Questa zona consente di immettere l'intervallo di attesa massimo tra i due segnali. Allo scadere di tale valore, la richiesta non viene trasmessa sul bus. I valori configurabili sono compresi tra 0 s e 10 s.

NOTA: L'impostazione predefinita è 0 ms.

NOTA: un valore di 0 s indica che il ritardo tra i due segnali non è stato gestito.

RTS/CTS Algoritmo del controllo di flusso

Lo scopo è di prevenire un overflow del buffer di ricezione.

Il segnale in uscita RTS di ciascuna periferica è collegato al segnale in ingresso CTS dell'altra periferica. Il trasmettitore (M340) è autorizzato a trasmettere i dati durante la ricezione del segnale in ingresso RTS (es. un altro M340) sul suo ingresso CTS . Questo algoritmo è simmetrico e consente la comunicazione asincrona full duplex.

Sezione 6.3
Programmazione comunicazione in modalità caratteri

Funzioni di comunicazione in modalità carattere

Funzioni disponibili

Per inviare e ricevere i dati tramite un canale di comunicazione in modalità carattere sono definite tre funzioni di comunicazione specifiche:

- PRINT_CHAR: invio di una stringa di caratteri di 1024 byte max.
- INPUT_CHAR: lettura di una stringa di caratteri di 1024 byte max.
- INPUT_BYTE *(vedi Unity Pro, Comunicazione, Libreria dei blocchi funzione)*: lettura di un array di byte di 1024 byte max.

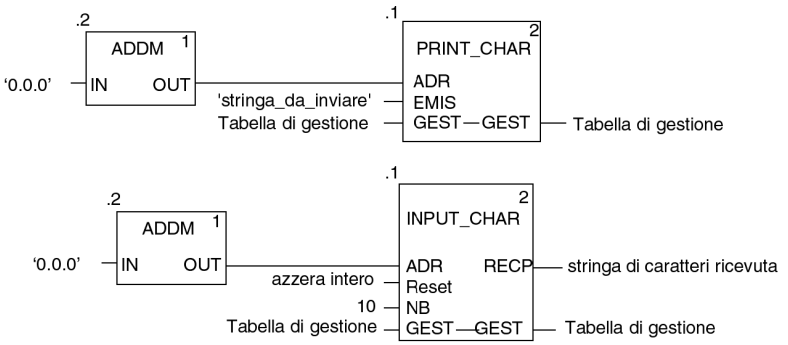
La porta seriale del PLC Modicon M340 è full duplex, quindi è possibile inviare una funzione PRINT_CHAR anche quando è stata inviata ed è ancora in attesa una funzione INPUT_CHAR.

NOTA: per la funzione INPUT_CHAR, è necessario configurare un time-out quando il canale è configurato senza Stop su Silenzio e ciò per il riconoscimento del bit di attività della funzione. Per la funzione PRINT_CHAR, è consigliabile ma non necessario configurare un time-out.

NOTA: A differenza del NOM0200 con un collegamento RS485, la CPU salva l'ECHO (replica) dei Dati trasmessi nello stesso buffer dei Dati ricevuti. Pertanto è obbligatorio azzerare il buffer della CPU dopo la funzione PRINT_CHAR o prima che vengano inviati dei Dati al canale. In caso contrario i dati ricevuti da un INPUT_CHAR o da INPUT_BYTE non saranno quelli previsti. Per azzerare il buffer della CPU è possibile eseguire un INPUT_CHAR con il buffer di Reset attivato e cancellare questo EF prima della scadenza del Timeout.

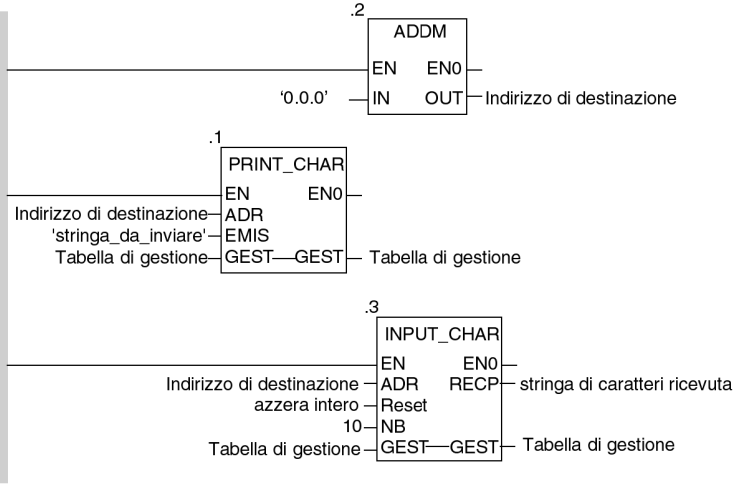
Esempio di programmazione in FBD

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione PRINT_CHAR e INPUT_CHAR nel linguaggio FBD:



Esempio di programmazione in Ladder

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione PRINT_CHAR e INPUT_CHAR nel linguaggio Ladder:



Esempio di programmazione in ST

Le righe di codice seguenti mostrano un esempio di programmazione delle funzioni di comunicazione `PRINT_CHAR` e `INPUT_CHAR` nel linguaggio ST:

```
PRINT_CHAR(ADDM('0.0.0'), 'stringa da inviare, Gestione tabella);
INPUT_CHAR(ADDM('0.0.0'), reimposta a 0, 10, Tabella di gestione, stringa
di caratteri ricevuta);
```

Caratteristica della funzione INPUT_CHAR

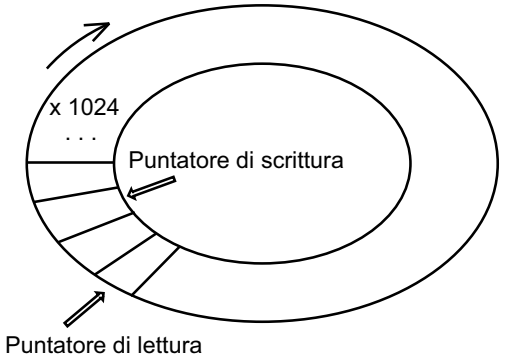
Se il parametro di ingresso Reset è impostato su 1, il buffer viene prima azzerato, quindi il processore attende la ricezione dei dati. L'uso di questa funzione è consigliato per avviare correttamente una ricezione rimuovendo i dati vecchi che possono rimanere nel buffer.

Meccanismo interno della CPU

I dati ricevuti vengono archiviati nel buffer ciclico da 1024 bit: una volta che il buffer è stato totalmente riempito, il 1025° bit ricevuto sovrascrive il 1° bit e così via. Ciascun bit del buffer letto attraverso la funzione INPUT CHAR viene azzerato.

Due puntatori indipendenti consentono accesso per la lettura e scrittura dei dati.

La figura seguente rappresenta questo meccanismo:



Annullamento di uno scambio

Due modi di programmazione consentono di annullare uno scambio eseguito dalle funzioni di comunicazione PRINT_CHAR e INPUT_CHAR. Essi sono presentati di seguito in linguaggio ST:

- Uso della funzione CANCEL:

```
IF (%MW40.0) THEN
 %MW200:=SHR(%MW40,8;)
 CANCEL(%MW200,%MW185);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione PRINT_CHAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma esegue le istruzioni seguenti:
  - Sposta i bit %MW40 di un byte (8 bit) a destra e carica il byte corrispondente al numero di scambio della comunicazione nella parola %MW200 .
  - annulla lo scambio il cui numero è contenuto nella parola %MW200 mediante la funzione CANCEL.

- Tramite il bit di annullamento della funzione di comunicazione:

```
IF (%MW40.0) THEN
 SET(%MW40.1);
 PRINT_CHAR(ADDM('0.0.0'), 'stringa da inviare, %MW40:4);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione PRINT_CHAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma imposta il bit %MW40.1, il bit di annullamento della funzione, a 1. Quest'azione interrompe la comunicazione della funzione PRINT_CHAR.

NOTA: quando si utilizza il bit di annullamento della funzione di comunicazione, occorre richiamare la funzione per attivare il bit di annullamento contenuto nella parola di gestione di scambio della funzione (%MW40 in questo esempio).

NOTA: quando si utilizza il bit di annullamento della funzione di comunicazione, è possibile annullare una comunicazione da una tabella di animazione impostando semplicemente il bit di annullamento della funzione a 1 (in questo esempio %MW40.1).

NOTA: questo esempio di programmazione è relativo alla funzione PRINT_CHAR, ma è applicabile anche alla funzione INPUT_CHAR.

NOTA: la funzione CANCEL utilizza una parola di rapporto (%MW185 in questo esempio).

Descrizione dei parametri della funzione ADDM

La tabella seguente descrive i diversi parametri della funzione ADDM:

Parametro	Tipo	Descrizione
IN	STRING	Indirizzo del dispositivo su bus o collegamento seriale. La sintassi dell'indirizzo è del tipo 'r.m.c.node'. L'indirizzo è composto dai seguenti parametri: ● **r**: numero di rack del sistema di destinazione, (sempre = 0) ● **m**: numero di slot del sistema di destinazione all'interno del rack, sempre = 0 ● **c**: numero di canale, sempre = 0 in quanto il collegamento seriale di un sistema remoto è sempre canale 0. ● **node**: campo opzionale che può essere SYS o vuoto.
OUT	ARRAY [0..7] OF INT	Tabella con l'indirizzo di un dispositivo. Questo parametro può essere usato come un parametro d'ingresso per diverse funzioni di comunicazione.

Descrizione dei parametri della funzione PRINT_CHAR

La tabella seguente descrive i diversi parametri della funzione PRINT_CHAR:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo del canale della modalità caratteri di ricezione del messaggio dato dal parametro OUT della funzione ADDM.
EMIS	STRING	Stringa di caratteri da inviare.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: numero di scambio
		  ● Byte meno significativo: bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: rapporto operazione
		  ● Byte meno significativo: rapporto comunicazione
		● Parola rango 3: una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: una parola gestita dall'utente, che definisce la lunghezza dello scambio.
		  ● Se la lunghezza di questo parametro è impostata a 0, il sistema invierà la stringa completa.
		  ● Se la lunghezza di questo parametro è superiore alla lunghezza della stringa, viene emesso l'errore 16#0A (Dimensione del buffer di invio insufficiente) nella seconda parola di gestione e non viene inviato alcun carattere.
		Nota: nel caso di PRINT_CHAR, la quarta parola di gestione (lunghezza da inviare) non deve essere scritta mentre il bit di attività è 1 (ossia, con l'EF è in esecuzione). Diversamente, PRINT_CHAR potrebbe venire bloccato con l'errore 0xB.

Descrizione dei parametri della funzione INPUT_CHAR

La tabella seguente descrive i diversi parametri della funzione `INPUT_CHAR`:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo del canale della modalità caratteri di ricezione del messaggio dato dal parametro OUT della funzione ADDM.
Reset	INT	Questo parametro può assumere due valori: ● Valore 1: reimposta memoria ricezione modulo a 0 ● Valore 0: non reimpostare memoria ricezione modulo a 0
NB	INT	Lunghezza della stringa di caratteri da estrarre.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: numero di scambio
		  ● Byte meno significativo: bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: rapporto operazione
		  ● Byte meno significativo: rapporto comunicazione
		● Parola rango 3: una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: una parola gestita dal sistema, che definisce la lunghezza dello scambio.

RECP	STRING	Stringa di caratteri ricevuta. Viene memorizzata in una stringa di caratteri.

Sezione 6.4
Debugging della comunicazione in modalità caratteri

Schermata di debug della comunicazione in modalità carattere

Generale

La schermata di debug per la modalità caratteri è accessibile solo in modalità online.

Accesso alla schermata di debug

La tabella seguente descrive la procedura per accedere alla schermata di debug della comunicazione in modalità caratteri.

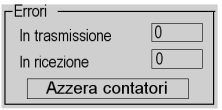
Passo	Azione
1	Accedere alla schermata di configurazione della comunicazione in modalità carattere. *(vedi pagina 107)*
2	Selezionare la scheda "Debug" sulla schermata visualizzata.

Descrizione della schermata Debug

La schermata di debug è composta dalla zona Errore e dalla zona Segnali.

La zona Errore

La zona Errore è illustrata di seguito:



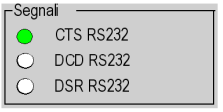
Questa zona indica il numero di interruzioni di comunicazione conteggiati dal processore.

- **In trasmissione**: corrisponde al numero di interruzioni verificatisi durante la trasmissione (immagina della parola `%MW4`).
- **In ricezione**: corrisponde al numero di interruzioni verificatisi durante la ricezione (immagina della parola `%MW5`).

Il pulsante Azzera contatori azzera tutti i contatori.

La zona Segnali

La zona Segnali è illustrata di seguito:



Questa zona indica l'attività dei segnali:

- **CTS RS232**: indica l'attività del segnale CTS.
- **DCD RS232**: non gestita dal processore (nessuna attività su questo LED).
- **DSR RS232**: non gestita dal processore (nessuna attività su questo LED).

Capitolo 7
Comunicazione Modbus Seriale per BMX NOM 0200

Oggetto del capitolo

Questo capitolo illustra il processo di implementazione software per la comunicazione Modbus seriale per BMX NOM 0200.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
7.1	Generalità	126
7.2	Configurazione della comunicazione Modbus Seriale	133
7.3	Programmazione della comunicazione Modbus Seriale	150
7.4	Debug della comunicazione Modbus Seriale	166

Sezione 7.1
Generalità

Oggetto della sezione

Questa sezione presenta i punti generali relativi alla comunicazione Modbus Seriale e ai relativi servizi.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Informazioni sul Modbus Seriale	127
Prestazioni	128
Accesso ai parametri del collegamento seriale	130

Informazioni sul Modbus Seriale

Introduzione

La comunicazione tramite Modbus permette lo scambio di dati tra tutte le periferiche connesse al bus. Il Modbus Seriale è un protocollo che crea un struttura gerarchica (un master e diversi slave).

Il master gestisce tutti gli scambi in due modi:

- il master scambia con lo slave e attende la risposta;
- il master scambia con tutti gli slave senza attendere risposta (trasmissione generale).

NOTA: Fare attenzione che i due master (sullo stesso bus) non inviino richieste simultaneamente, altrimenti le richieste vanno perdute e ogni report avrà un risultato errato che potrebbe essere 16#0100 (impossibile elaborare la richiesta) o 16#ODFF (slave non presente).

⚠ AVVERTENZA
PERDITA CRITICA DEI DATI La porta di comunicazione deve essere usata solo per il trasferimento di dati non critici. **Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.**

Prestazioni

In breve

È possibile utilizzare le tabelle seguenti per valutare i tempi di scambio tipici della comunicazione Modbus in base a criteri diversi.

I risultati visualizzati corrispondono al periodo operativo medio della funzione `READ_VAR` in millisecondi.

Definizione di tempo di scambio

Il Tempo di scambio è il tempo che intercorre tra l'attivazione e la fine dello scambio. Questo valore include il tempo di comunicazione del collegamento seriale.

Lo scambio è creato quando si richiama la funzione di comunicazione.

Lo scambio termina quando si verifica uno dei seguenti eventi:
- i dati vengono ricevuti
- si verifica un'anomalia
- il timeout scade

Tempo di scambio per una Parola

Nella seguente tabella sono indicati i tempi di scambio per una parola di comunicazione Modbus su un modulo BMX NOM 0200:

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Durata dello scambio in ms Lo slave Modbus è un BMX P34 1000 ciclico
4800	Ciclica	65
4800	10	68
4800	50	100
9600	Ciclico	38
9600	10	47
9600	50	50
19200	Ciclico	29
19200	10	38
19200	50	50
38400	Ciclico	24
38400	10	30
38400	50	50
57600	Ciclico	17
57600	10	20

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Durata dello scambio in ms Lo slave Modbus è un BMX P34 1000 ciclico
57600	50	50
115200	Ciclico	17
115200	10	20
115200	50	50

Tempo di scambio per 100 parole

Nella seguente tabella sono indicati i tempi di scambio per 100 parole di comunicazione Modbus su un processore BMX NOM 0200:

Velocità di trasmissione della comunicazione in bit al secondo	Durata del ciclo in ms	Tempo di scambio in ms Lo slave Modbus è un BMX P34 1000 ciclico
4800	Ciclico	560
4800	10	560
4800	50	600
9600	Ciclico	286
9600	10	295
9600	50	300
19200	Ciclico	152
19200	10	160
19200	50	200
38400	Ciclico	86
38400	10	90
38400	50	100
57600	Ciclico	56
57600	10	60
57600	50	100
115200	Ciclico	36
115200	10	40
115200	50	50

Precisione della misura

Tutti i tempi di scambio indicati in precedenza derivano da misurazioni eseguite con un margine di precisione di +/-10 ms.

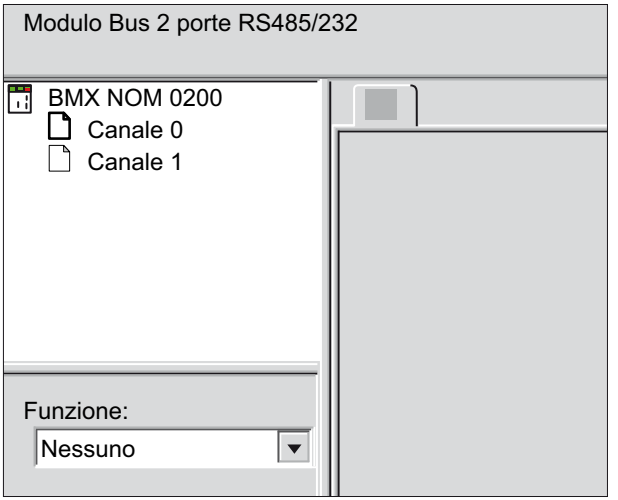
Accesso ai parametri del collegamento seriale

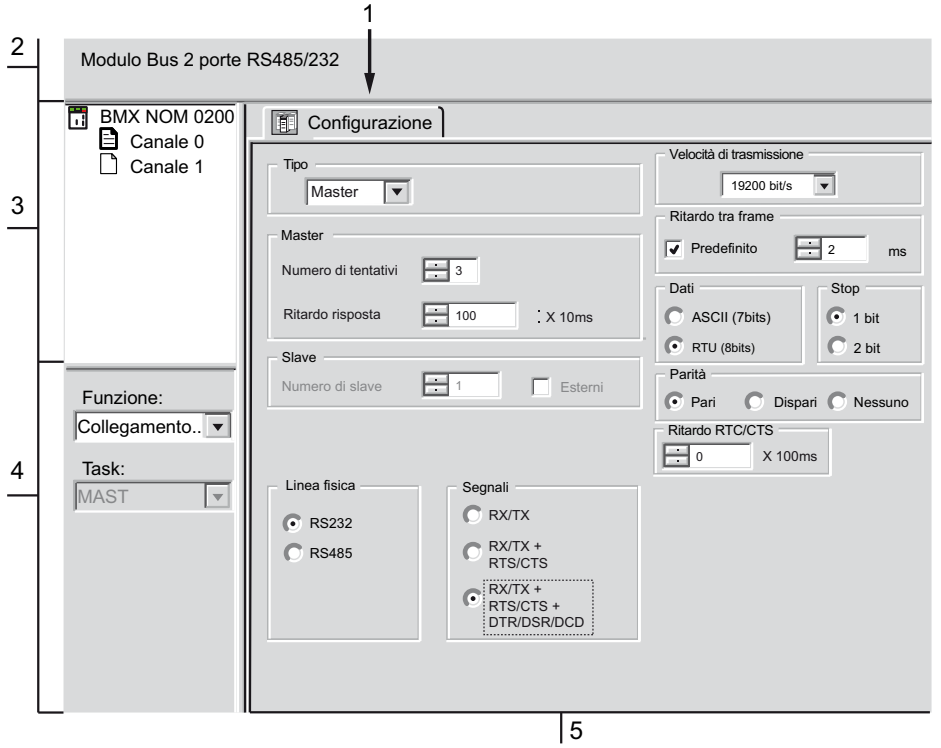
In breve

Le sezioni che seguono descrivono l'accesso alla schermata di configurazione della porta seriale del modulo BMX NOM 0200 e gli elementi generali delle schermate di configurazione e di debug del collegamento Modbus e Modalità caratteri:

Accesso al collegamento seriale

La tabella seguente descrive la procedura per accedere al collegamento seriale di un modulo BMX NOM 0200:

Passo	Azione
1	Aprire l'editor di configurazione hardware.
2	Fare doppio clic sul modulo BMX NOM 0200.
3	Selezionare il canale da configurare (Canale 0 o Canale 1). **Risultato con Canale 0 selezionato:** 

Passo	Azione
4	Selezionare la funzione **collegamento Modbus** . **Risultato con Canale 0 selezionato:** 

Descrizione della schermata Configurazione

La seguente tabella mostra gli elementi delle schermate di configurazione:

Legenda	Elemento	Funzione
1	Schede	La scheda in primo piano indica la modalità correntemente in uso (**Configurazione** in questo esempio). Ogni modalità può essere selezionata dalla scheda corrispondente. Le modalità disponibili sono: ● **Configurazione** ● **Debug** (accessibile solo in modalità online) ● **Diagnostica** (accessibile solo in modalità online)
2	Area modulo	Visualizza lo stato dei LED di riferimento e del modulo in modalità online.

3	Zona canale	Consente di:
		● Visualizzare le seguenti schede facendo clic su **BMX NOM 0200**:
		  ● "Panoramica", che mostra le caratteristiche del dispositivo
		  ● **"Oggetti di I/O"** *(vedi Unity Pro, Modalità operative)*, utilizzato per presimbolizzare gli oggetti di ingresso/uscita
		  ● "Errore", che riporta eventuali errori del dispositivo (in modalità online).
		● Visualizzare le seguenti schede facendo clic su **Canale 0** o **Channel 1**:
		  ● "Configurazione"
		  ● "Debug"
		  ● "Errore"
		● Visualizzare il simbolo e il nome del canale definito dall'utente (utilizzando l'editor delle variabili).

4	Zona parametri generali	Consente di selezionare i parametri generali associati al canale: ● **Funzione**: Le funzioni disponibili sono"Nessuna", "collegamento Modbus" e "modalità carattere". Di default, la funzione "Nessuna" è configurazione. ● **Task**: Definisce il task master in cui verranno scambiati gli oggetti di scambio impliciti del canale. Quest'area è grigiata e non può essere configurata.
5	Area di configurazione, debug o errore	In modalità configurazione, viene usata per configurare i parametri del canale. In modalità debug, viene usata per eseguire il debug del canale di comunicazione. In modalità diagnostica, viene usata per visualizzare gli errori rilevati a livello del modulo o del canale.

Sezione 7.2
Configurazione della comunicazione Modbus Seriale

Oggetto della sezione

Questa sezione descrive il processo di configurazione software per la comunicazione Modbus Seriale.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Schermata di configurazione della comunicazione seriale Modbus in un rack locale Modicon M340	134
Schermata di configurazione per la comunicazione seriale Modbus del modulo BMX NOM 0200 in derivazione X80	137
Funzioni accesibili Modbus	140
Valori predefiniti dei parametri di comunicazione Modbus Seriale	141
Parametri Modbus legati all'applicazione	142
Parametri Modbus legati alla trasmissione	144
Parametri del segnale e della linea fisica in Modbus	146
Come impostare l'indirizzo slave Modbus del BMX NOM0200 senza Unity Pro?	148

Schermata di configurazione della comunicazione seriale Modbus in un rack locale Modicon M340

Informazioni generali

Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modbus seriale.

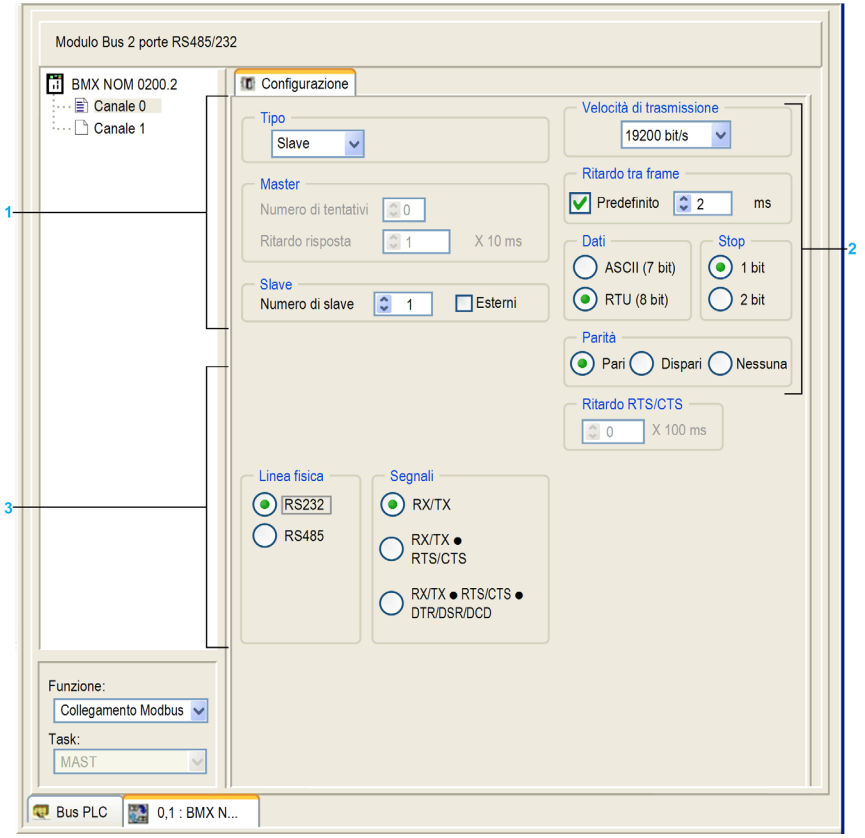
Accesso alla schermata di configurazione

La tabella seguente descrive la procedura per accedere alla schermata di configurazione della comunicazione Modbus seriale:

Passo	Azione
1	Aprire la sottodirectory BMX NOM 0200 nel browser di progetto *(vedi pagina 130)*.
2	Selezionare il Canale da configurare e la funzione "collegamento Modbus" sullo schermo che compare.

Illustrazione

La figura di seguito mostra la schermata di configurazione predefinita per la comunicazione Modbus seriale sul Canale 0:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione del collegamento Modbus:

Legenda	Elemento	Commento
1	Parametri dell'applicazione *(vedi pagina 142)*	Tali parametri sono accessibili da tre zone: ● Tipo, ● Master, ● Slave.
2	Parametri di trasmissione *(vedi pagina 144)*	Tali parametri sono accessibili da cinque zone: ● Velocità di trasmissione, ● Ritardo tra frame, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 146)*	Tali parametri sono accessibili da tre zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS.

NOTA: Durante la configurazione della comunicazione Modbus Seriale in modalità Master, la zona "Slave" è disattivata e non può essere modificata e viceversa.

Schermata di configurazione per la comunicazione seriale Modbus del modulo BMX NOM 0200 in derivazione X80

Informazioni generali

Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modbus seriale.

NOTA: quando un BMX NOM 0200 si trova in una derivazione Quantum Ethernet I/O X80, deve avere:

- Software prodotto = 04
- Software software ≥ 1.4

Questa informazione è visibile sull'etichetta posta sul lato del modulo.

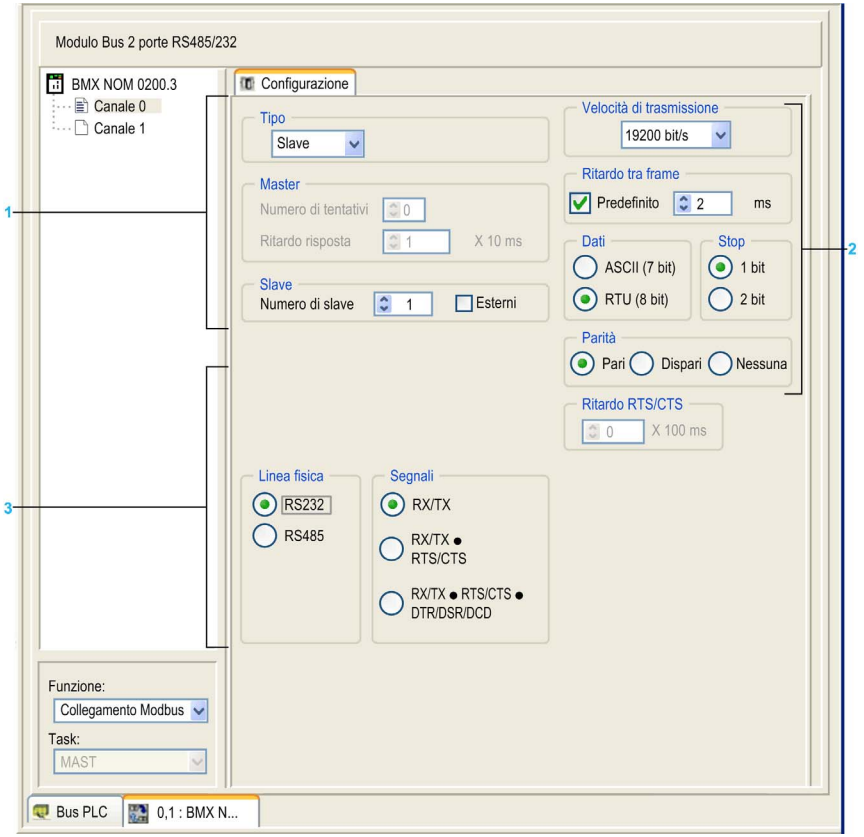
Accesso alla schermata di configurazione

La tabella seguente descrive la procedura per accedere alla schermata di configurazione della comunicazione Modbus seriale:

Passo	Azione
1	Aprire la sottodirectory BMX NOM 0200 nel browser di progetto *(vedi pagina 130)*.
2	Selezionare il Canale da configurare e la funzione "collegamento Modbus" sullo schermo che compare.

Figura

La figura di seguito mostra la schermata di configurazione predefinita per la comunicazione Modbus seriale sul Canale 0:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione del collegamento Modbus:

Legenda	Elemento	Commento
1	Parametri dell'applicazione *(vedi pagina 142)*	Tali parametri sono accessibili da tre zone: ● Tipo, ● Master, ● Slave.
2	Parametri di trasmissione *(vedi pagina 144)*	Tali parametri sono accessibili da cinque zone: ● Velocità di trasmissione, ● Ritardo tra frame, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 146)*	Tali parametri sono accessibili da tre zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS.

NOTA: Quando si configura la comunicazione seriale Modbus in modalità Master, i parametri **Slave** sono visualizzati in grigio e non possono essere modificati.

Funzioni accesibili Modbus

In breve

L'accessibilità delle funzioni per la configurazione del collegamento seriale di un modulo BMX NOM 0200 tramite il protocollo Modbus seriale dipende dal collegamento fisico utilizzato.

Funzioni accessibili

La tabella seguente mostra le varie funzioni configurabili in base al tipo di collegamento seriale utilizzato.

Funzione	Collegamento RS485 (su Canale 0 o Canale 1)	Collegamento RS232 (su Canale 0)
Numero tentativi master	X	X
Ritardo risposta master	X	X
Numero di slave	X	X
Velocità di trasmissione	X	X
Ritardo tra frame	X	X
Dati	• ASCII (7 bit) • RTU (8 bit)	• ASCII (7 bit) • RTU (8 bit)
Stop	• 1 bit • 2 bit	• 1 bit • 2 bit
Parità	• Odd • Pari • Nessuno	• Odd • Pari • Nessuno
Segnali RX/TX	X	X
Segnali RTS/CTS	-	X
Ritardo RTS/CTS	-	X
Segnali DTR/DSR/DCD	-	X
Polarizzazione	-	-

X Funzione accessibile
- Funzione inaccessibile

Valori predefiniti dei parametri di comunicazione Modbus Seriale

In breve

Tutti i parametri di comunicazione Modbus Seriale presentano valori predefiniti.

Valori predefiniti

La tabella di seguito mostra i valori predefiniti per i parametri di comunicazione seriale Modbus sul Canale 0 e il Canale 1 del modulo BMX NOM 0200:

Parametro di configurazione	**esad.**
Modalità	Slave
Linea fisica	RS232
Numero di slave	1
Ritardo tra frame	2 ms
Velocità di trasmissione	19200 bit/s
Parità	Pari
Bit di dati	RTU (8 bit)
Bit di stop	1 bit

Parametri Modbus legati all'applicazione

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri dell'applicazione.

Tali parametri sono accessibili da tre zone di configurazione:
- La zona Tipo,
- La zona Master,
- La zona Slave.

La zona Tipo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di selezionare il ruolo da configurare per il modulo nella comunicazione Modbus seriale:
- **Master**: Quando il modulo è il master.
- **Slave**: Quando il modulo è uno slave.

La zona Master

La zona di configurazione illustrata di seguito è accessibili solo se si seleziona "Master" nella zona "Tipo":

Questa zona consente di specificare i parametri seguenti:
- **Numero di tentativi**: numero di tentativi di connessione eseguiti dal master prima di considerare lo slave assente.
  il valore predefinito è 3;
  intervallo di valori possibili da 1 a 15;
  un valore di 0 indica nessun tentativo da parte del master.
- **Ritardo risposta**: il tempo tra la richiesta iniziale del master e un tentativo ripetuto in assenza di risposta dello slave. È il tempo massimo tra la trasmissione dell'ultimo carattere della richiesta del master e la ricezione del primo carattere della richiesta inviato dallo slave.
  Il valore predefinito è 1 secondo (100*10ms).
  intervallo di valori possibili da 10 ms a 10 s.

NOTA: Il ritardo della Risposta del Master deve essere almeno uguale al tempo di ritardo alla Risposta più lungo degli Slaves presenti sul bus.

NOTA: Nella modalità broadcast, il valore configurato come Ritardo risposta viene utilizzato come Ritardo broadcast: tempo minimo tra due scambi nella modalità broadcast.

La zona Slave

La zona di configurazione illustrata di seguito è accessibili solo se si seleziona "Slave" nella zona "Tipo":

Questa zona consente di specificare il numero di slave del processore:

il valore predefinito è 1;

intervallo di valori possibili da 1 a 247.

Selezionando **Esterno** si disattiva il campo **Numero dello Slave** e forza il modulo ad utilizzare il valore dell'indirizzo slave salvato nella *(vedi pagina 148)*memoria FLASH interna.

NOTA: Se l'indirizzo archiviato nella FLASH non è nell'indirizzo dell'intervallo MODBUS, allora verrà utilizzato l'indirizzo dello slave predefinito 248.

Quando il firmware del modulo viene aggiornato, l'indirizzo predefinito dello slave archiviato nella FLASH viene impostato su 248. Deve essere usato un nuovo comando per rinizializzare l'indirizzo FLASH.

Parametri Modbus legati alla trasmissione

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di trasmissione.

Tali parametri sono accessibili da cinque zone:
- La zona Velocità di trasmissione
- La zona Ritardo tra caratteri;
- La zona Dati,
- La zona Stop,
- La zona Parità.

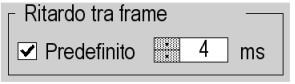
La zona Velocità di trasmissione

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

È possibile utilizzarla per selezionare la velocità di trasmissione del collegamento Modbus Seriale. La velocità selezionata deve essere coerente con gli altri dispositivi. I valori configurabili sono 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 e 115200 (solo sul canale 0 in modalità RS232) bit al secondo.

L'area Ritardo tra frame

La zona di configurazione illustrata di seguito è accessibile solo in modalità RTU (viene disattivata in modalità ASCII):



Il Ritardo tra frame è l'intervallo minimo di separazione di ricezione di due frame. Tale ritardo viene gestito quando il BMX NOM 0200 (master o slave) sta ricevendo dei messaggi.

NOTA: il valore predefinito dipende dalla velocità di trasmissione selezionata.

NOTA: Il ritardo tra le frame dovrebbe essere rappresentato dal valore Predefinito in modo da essere conforme al Modbus. Nel caso in cui uno Slave non sia conforme, il valore può essere cambiato e deve essere identico sia per Master che per tutti gli Slave del Bus.

La zona Dati

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di specificare il tipo di codifica utilizzato per comunicare con il collegamento Modbus Seriale. Questo campo è impostato in base agli altri dispositivi collegati sul bus. Sono possibili due modalità di configurazione:

- Modalità RTU:
  - i caratteri sono codificati su 8 bit;
  - la fine del frame viene individuata in presenza di un silenzio di almeno 3,5 caratteri;
  - L'integrità del frame viene controllata tramite una parola nota come checksum CRC, che è contenuta nel frame.
- modalità ASCII:
  - i caratteri sono codificati su 7 bit;
  - l'inizio del frame viene individuato alla ricezione del carattere **":"** ;
  - la fine del frame è individuata da un ritorno a capo e da un avanzamento riga;
  - l'integrità del frame viene controllata tramite un byte noto come checksum LRC, che è contenuto nel frame.

La zona Stop

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

La zona Stop consente di inserire il numero di bit di stop utilizzati per la comunicazione. Questo campo viene definito in base agli altri dispositivi. I valori configurabili sono:

- 1 bit
- 2 bit

La zona Parità

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Queste zone consentono di stabilire l'aggiunta o meno di un bit di parità e di definirne il tipo. Questo campo viene definito in base agli altri dispositivi. I valori configurabili sono:

- Pari
- Odd
- Nessuno

Parametri del segnale e della linea fisica in Modbus

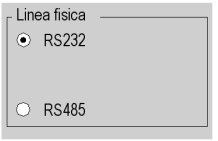
In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di segnale e linea fisica.

Tali parametri sono accessibili da tre zone:
- La zona Linea fisica,
- La zona Segnali,
- la zona Ritardo RTS/CTS.

La zona Linea fisica

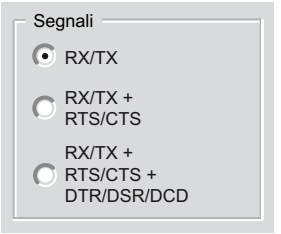
La zona di configurazione illustrata di seguito è accessibile solo sul Canale 0 (viene disattivata e configurata su RS485 sul Canale 1):



In questa zona, è possibile scegliere due tipi di linea fisica per la porta seriale sul modulo BMX NOM 0200:
- La linea RS232,
- La linea RS485,

La zona Segnali

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



In questa zona, è possibile selezionare i segnali supportati dalla linea fisica RS232:
- RX/TX
- RX/TX + RTS/CTS (segnali di gestione del flusso hardware)
- RX/TX + RTS/CTS + DTR/DSR/DCD (Segnali modem)

Se è configurata la linea RS485, tutta la zona sarà disattivata e il valore predefinito sarà RX/TX.

La RTS/CTS zona Ritardo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

La RTS/CTS zona ritardo è disponibile solamente quando entrambe le caselle di selezione RS232 e RX/TX+RTS/CTS o RX/TX+RTS/CTS+DTR/DSR/DCD sono selezionate. Viene eseguito un RTS/CTS controllo del flusso hardware.

L'algoritmo RTS/CTS di controllo del flusso hardware ha lo scopo di prevenire il buffer di ricezione dell'overflow (full duplex).

Il RTS/CTS ritardo corrisponde al ritardo del time out tra l'innalzamento RTS e l'innalzamento CTS. Un valore RTS/CTS di ritardo diverso da 0 inoltre corrisponde al tempo massimo di attesa tra ciascuna trasmissione di caratteri dopo l'innalzamento dei segnali RTS e CTS. Se il valore è impostato sullo 0, gli UART possono rimanere bloccati in uno stato di attesa per un tempo infinito finché incrementa il CTS, e il valore 0 viene quindi utilizzato solamente in casi particolari, ad esempio, per collegare ad anello il segnale RTS con il segnale CTS e verificare che nessun cavo di collegamento sia difettoso.

NOTA: L'impostazione predefinita è 0 ms.

Come impostare l'indirizzo slave Modbus del BMX NOM0200 senza Unity Pro?

Condizioni e prerequisiti

L'indirizzo FLASH può essere aggiornato in qualunque modalità, ma viene preso in considerazione solo quando viene eseguita una `modalità operativa`.

L'elenco di seguito indica le condizioni e i prerequisiti per impostare l'indirizzo MODBUS del BMX NOM0200 senza Unity Pro:

- Per usare l'indirizzo FLASH, il modulo deve essere configurato:
  - con il protocollo slave MODBUS con la casella di controllo su **EXTERNAL**.
  - con il protocollo master MODBUS o in modalità **CHAR** e poi passato sul protocollo slave MODBUS.

Aggiornare l'indirizzo slave MODBUS nella FLASH tramite i comandi Applicative

La tabella che segue indica le operazioni per aggiornare l'indirizzo slave MODBUS nella FLASH con i comandi Applicative:

Passo	Azione
1	Memorizzare l'indirizzo slave in `%MWr.m.c.25`.
2	Impostare il bit `%MWr.m.c.24.7`.
3	Inviare il comando `WRITE_CMD` al canale del modulo.
4	Controllare la fine del comando (`%MWr.m.c.0.1` non riuscito) e se il comando è accettato (se `%MWr.m.c.1.1` è a zero significa che non ci sono errori) => la FLASH è aggiornata.
5	Eseguire una delle seguenti modalità operative sul canale per prendere in considerazione il nuovo indirizzo: • Scaricamento di un'applicazione • Avvio a freddo • Riavvio a caldo • Sostituzione a caldo • Cambio di protocollo (TO SLAVE)
6	Eseguire un READ_STS sul canale per verificare l'indirizzo slave nel byte più significativo di `%MWr.m.c.3`.

NOTA: Diversi ordini possono essere integrati nello stesso comando. Se uno degli ordini non può essere eseguito, l'intero comando verrà rifiutato e nessun ordine verrà eseguito.

Aggiornare l’indirizzo slave MODBUS nella FLASH su una linea seriale

La tabella che segue indica le operazioni per aggiornare l’indirizzo slave MODBUS nella FLASH su una linea seriale:

Passo	Azione
1	Configurare l’apparecchiatura MASTER con lo stesso parametro della linea seriale che dispone un canale del modulo.
2	Collegare il MASTER al modulo con una connessione punto punto.
3	Inviare la richiesta `0x11` all’indirizzo punto punto: `0xF8 0x11 0x01` numero canale (0 o 1) slaveID(`0..0xF8`)
4	Verificare che la risposta sia OK => la FLASH è aggiornata.
5	Eseguire una modalità operativa sul canale per prendere in considerazione le modifiche apportate al passo 4.
6	Inviare una richiesta `0x11` per verificare il nuovo indirizzo slave: slaveID `0x11 0x01`

NOTA: Non modificare la FLASH regolarmente per evitare di danneggiare questo componente (100.000 cicli di scrittura max).

Sezione 7.3
Programmazione della comunicazione Modbus Seriale

Oggetto della sezione

Questa sezione descrive il processo di programmazione utilizzato per implementare la comunicazione Modbus seriale.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Servizi supportati da un modulo master del collegamento Modbus	151
Servizi supportati da un modulo slave del collegamento Modbus	158
Informazioni dettagliate sulla modalità Expert Modbus	160

Servizi supportati da un modulo master del collegamento Modbus

In breve

Se utilizzato come master in un collegamento Modbus, un modulo BMX NOM 0200 supporta diversi servizi tramite le funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH .

Scambi di dati

La lettura o scrittura delle variabili viene effettuata indirizzando le seguenti richieste al dispositivo slave di destinazione.

Tale richieste utilizzano le funzioni di comunicazione READ_VAR e WRITE_VAR.

Richiesta Modbus	Codice funzione	Funzione di comunicazione
Lettura bit	16#01 o 16#02	READ_VAR
Lettura parole	16#03 o 16#04	READ_VAR
Scrittura bit	16#0F	WRITE_VAR
Scrittura parole	16#10	WRITE_VAR

Più in generale, è possibile inviare qualsiasi richiesta Modbus a un dispositivo slave tramite la funzione di comunicazione DATA_EXCH .

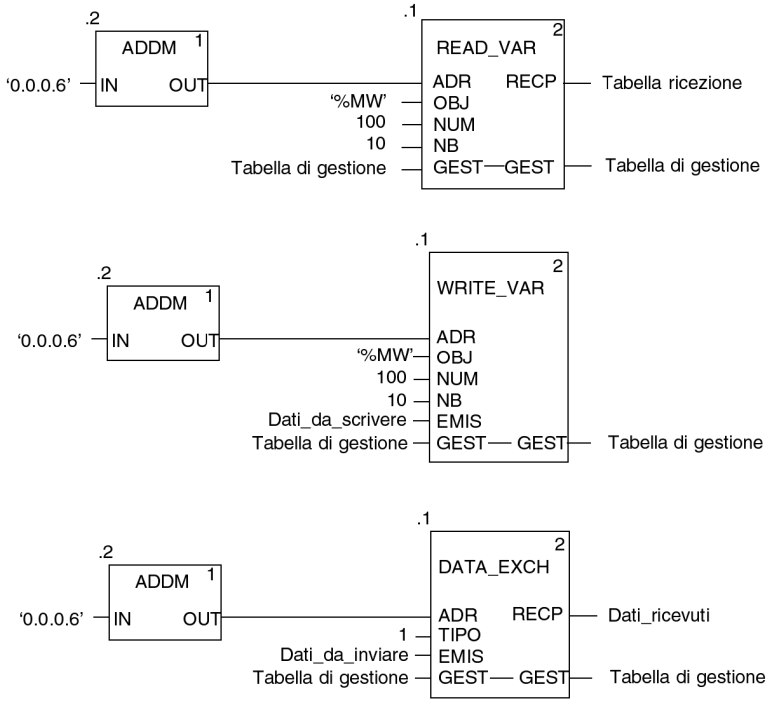
Funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH

Per inviare e ricevere i dati tramite un canale di comunicazione Modbus sono definite tre funzioni di comunicazione specifiche:

- READ_VAR: Per leggere le variabili
- WRITE_VAR: Per scrivere le variabili
- DATA_EXCH: Per inviare le richieste Modbus a un altro dispositivo sul protocollo selezionato

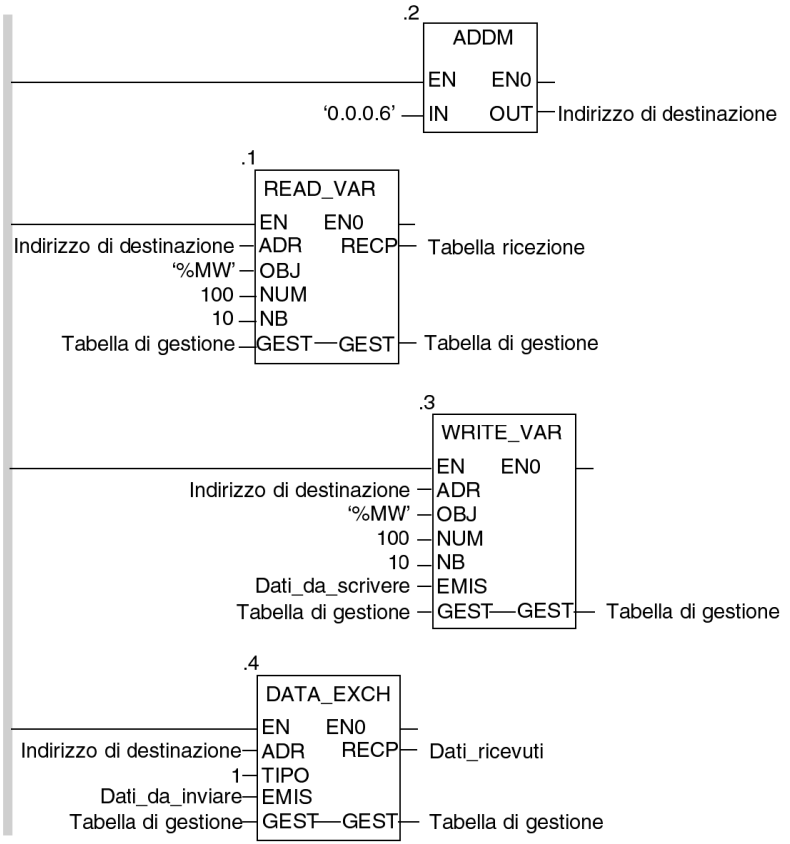
Esempio Programmazione in FBD

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio FBD :



Esempio Programmazione in Ladder

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio Ladder:



Esempio di programmazione in ST

Le righe di codice di seguito rappresentano un esempio di programmazione delle funzioni di comunicazione READ_VAR, WRITE_VAR e DATA_EXCH nel linguaggio ST :

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,
Receiving_Table);
WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,
Management_Table);
DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,
Received_data);
```

Annullamento di uno scambio

Uno scambio eseguito dalle funzioni READ_VAR, WRITE_VAR e DATA_EXCH può essere annullato secondo due modi di programmazione, presentati in linguaggio ST di seguito:

- Uso della funzione CANCEL:

```
IF (%MW40.0) THEN
 %MW200:=SHR(%MW40,8;)
 CANCEL(%MW200,%MW185);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione READ_VAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma esegue le istruzioni seguenti:
  - Sposta i bit %MW40 di un byte (8 bit) a destra e carica il byte corrispondente al numero di scambio della comunicazione nella parola %MW200 ,
  - annulla lo scambio il cui numero è contenuto nella parola %MW200 mediante la funzione CANCEL.

- Tramite il bit di annullamento della funzione di comunicazione:

```
IF (%MW40.0) THEN
 SET(%MW40.1);
 READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4, %MW10:10);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione READ_VAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma imposta il bit %MW40.1, il bit di annullamento della funzione, a 1. Quest'azione interrompe la comunicazione della funzione READ_VAR.

NOTA: Quando si utilizza il bit di annullamento della funzione di comunicazione contenuto nella parola di gestione scambio (%MW40, in questo esempio), deve essere chiamata la funzione (READ_VAR, in questo esempio) per poter attivare l'annullamento dello scambio.

NOTA: quando si utilizza il bit di annullamento della funzione di comunicazione, è possibile annullare una comunicazione da una tabella di animazione Questo può essere fatto semplicemente impostando il bit di annullamento funzione a 1 (%MW40.1, in questo esempio) e poi riavviare di nuovo la funzione di comunicazione.

NOTA: Questo esempio di programmazione è relativo alla funzione READ_VAR , ma è applicabile anche alla funzione WRITE_VAR e alle funzioni DATA_EXCH .

NOTA: la funzione CANCEL utilizza una parola di rapporto (%MW185 in questo esempio).

Descrizione dei parametri della funzione ADDM

La tabella seguente descrive i diversi parametri della funzione ADDM:

Parametro	Tipo	Descrizione
IN	STRING	Indirizzo del dispositivo su bus o collegamento seriale. La sintassi dell'indirizzo è del tipo 'r.m.c.node'. L'indirizzo è composto dai seguenti parametri: ● **r**: Numero di rack del modulo ● **m**: Numero di slot del modulo all'interno del rack ● **c**: Numero di canale del modulo ● **node**: Numero di slave a cui è stata inviata la richiesta.
OUT	ARRAY [0..7] OF INT	Array che rappresenta l'indirizzo di un dispositivo. Questo parametro può essere usato come un parametro d'ingresso per diverse funzioni di comunicazione.

Descrizione dei parametri della funzione READ_VAR

La tabella seguente descrive i diversi parametri della funzione READ_VAR:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
OBJ	STRING	Tipo di oggetto da leggere: I tipi disponibili sono indicate di seguito: ● %M: bit interno ● %MW: parola interna ● %I: bit di ingresso esterno ● %IW: parola di ingresso esterna
NUM	DINT	Indirizzo del primo oggetto da leggere:
NB	INT	Numero di oggetti consecutivi da leggere.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio,
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

RECP	ARRAY [n..m] OF INT	Tabella di parole contente il valore degli oggetti letti.

Descrizione dei parametri della funzione WRITE_VAR

La tabella seguente descrive i diversi parametri della funzione WRITE_VAR:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
OBJ	STRING	Tipo dell'oggetto da scrivere: I tipi disponibili sono indicate di seguito: ● %M: bit interno ● %MW: parola interna **Nota:** WRITE_VAR non può essere utilizzato per le variabili %I e %IW.
NUM	DINT	Indirizzo del primo oggetto da scrivere.
NB	INT	Numero di oggetti consecutivi da scrivere:
EMIS	ARRAY [n..m] OF INT	Tabella di parole contente il valore degli oggetti da scrivere.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio,
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

Descrizione dei parametri della funzione DATA_EXCH

La tabella seguente descrive i diversi parametri della funzione DATA_EXCH:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo dell'entità di destinazione dato dal parametro OUT della funzione ADDM.
TIPO	INT	Per i PLC Modicon M340, l'unico valore possibile è **1**: Trasmissione di un array **EMIS** , quindi il PLC attende la ricezione di un array **RECP** .
EMIS	ARRAY [n..m] OF INT	Tabella degli integrali da inviare al dispositivo di destinazione della richiesta. **Nota:** È imperativo che la lunghezza dei dati da inviare (byte) sia assegnata alla quarta parola della tabella di gestione prima di lanciare la funzione, affinché questa sia eseguita correttamente.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola rango 1: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Numero di scambio.
		  ● Byte meno significativo: Bit attività (rango 0) e bit annullamento (rango 1)
		● Parola rango 2: Una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: Rapporto operazione,
		  ● Byte meno significativo: Rapporto comunicazione.
		● Parola rango 3: Una parola gestita dall'utente, che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola rango 4: Una parola gestita dal sistema, che definisce la lunghezza dello scambio.

RECP	ARRAY [n..m] OF INT	Tabella degli integrali contenente i dati ricevuti. **Nota:** La dimensione dei dati ricevuti (in byte) viene scritta automaticamente dal sistema nella quarta parola della tabella di gestione.

Servizi supportati da un modulo slave del collegamento Modbus

In breve

Se utilizzato come slave in un collegamento Modbus, un modulo BMX NOM 0200 supporta diversi servizi.

Scambi di dati

Un modulo slave gestisce le richieste seguenti:

Richiesta Modbus	Codice funzione	Oggetto PLC
Lettura di n bit di uscita	16#01	%M
Lettura di n parole di uscita	16#03	%MW
Scrittura di n bit di uscita	16#0F	%M
Scrittura di n parole di uscita	16#10	%MW
Lettura/Scrittura di n parole di uscita	16#17	%MW

NOTA: Lettura/Scrittura di multiple %MW

`WRITE` esegue prima `READ` per poter poi scrivere e leggere gli stessi registri nello stesso momento dello `IOscanning`. Se la dimensione dello scambio di `WRITE` o di `READ` è fuori limite lo stato restituito sarà “ILLEGAL DATA ADDRESS”, ma se solo `READ` fallisce, verrà eseguito `WRITE` con lo stesso stato.

Diagnostica e manutenzione

Le richieste di diagnostica e manutenzione gestite da un modulo Modbus slave BMX NOM 0200 sono elencate di seguito:

Designazione	Codice funzione/Codice funzione secondaria
Leggere stato di eccezione	16#07
Riavviare opzione di comunicazione	16#08 / 16#01
Restituire registro diagnostica	16#08 / 16#02
Modificare delimitatore di ingresso ASCII	16#08 / 16#03
Forzare modalità solo ascolto	16#08 / 16#04
Eliminare contatori e registro diagnostica	16#08 / 16#0A
Restituire conteggio messaggio bus	16#08 / 16#0B
Restituire conteggio errore di comunicazione bus	16#08 / 16#0C
Restituire conteggio errore eccezione bus	16#08 / 16#0D
Restituire conteggio messaggio slave	16#08 / 16#0E

Designazione	Codice funzione/Codice funzione secondaria
Restituire conteggio nessuna risposta slave	16#08 / 16#0F
Restituire conteggio riconoscimenti negativi slave	16#08 / 16#10
Restituire conteggio occupato slave	16#08 / 16#11
Restituire conteggio overrun caratteri bus	16#08 / 16#12
Ottenere contatore eventi di comunicazione	16#0B
Ottenere log eventi di comunicazione	16#0C
Segnalare identificazione slave	16#11
Scrivere identificazione slave	16#11 / 16#01

Informazioni dettagliate sulla modalità Expert Modbus

Comunicazione in modalità Expert

La modalità Expert è un insieme di comandi che possono essere inviati al modulo per ottenere funzionalità aggiuntive.

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.24	CONTROL	Esplicito	INT	Segnale di comando, cambia protocollo
%MWr.m.c.24.0		Esplicito	BOOL	Cancella i contatori locali
%MWr.m.c.24.1		Esplicito	BOOL	Modifica dinamicamente il conteggio dei tentativi in modalità master MODBUS (%MW26)
%MWr.m.c.24.2		Esplicito	BOOL	Modifica il ritardo di risposta dello slave (%MW28) per uno slave specifico (%MW27) in modalità master
%MWr.m.c.24.3		Esplicito	BOOL	Modifica il tempo morto predefinito dello slave, lo slave ignora i caratteri ricevuti dopo la ricezione di un frame inoltrato alla CPU (%MW29)
%MWr.m.c.24.4		Esplicito	BOOL	Modifica la temporizzazione interna RTU MODBUS t1,5ch (%MW31), t3,5ch (%MW30) e il ritardo tra gli scambi (%MW32). L'aggiornamento di questi valori può creare interferenze se il modulo è in funzione
%MWr.m.c.24.6		Esplicito	BOOL	Cambia la modalità di gestione modem HALF/FULL DUPLEX ● Se è impostato simultaneamente con RTS_ON (%MWr.m.c.24.10 funziona anche con RTS_OFF %MWr.m.c.24.11 e usa il DTR se è utilizzato %MWr.m.r.24.8 o %MWr.m.r.24.9) è attivata la modalità modem half duplex. ● Se è impostato questo bit, ma nessuno dei segnali RTS/DTR (né %MWr.m.c.24.8, né %MWr.m.c.24.9, né %MWr.m.c.24.10, né %MWr.m.c.24.11) è attivata la modalità full duplex Il bit %MW26 è utilizzato per impostare l'inizio ritardo e %MW27 è utilizzato per impostare la fine ritardo. Pertanto, i bit %MW24.5, %MW24.1 e %MW24.2 non possono essere utilizzati simultaneamente. **NOTA**: è possibile che l'utente debba ripristinare lo stato corretto dei segnali RTS/DTR dopo l'accettazione del comando.

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.24.7	SAVE_SLAVE_ADDR	Esplicito	BOOL	Salva l'indirizzo slave Modbus nella FLASH (%MW25).
%MWr.m.c.24.8	DTR_ON	Esplicito	BOOL	Imposta il segnale DTR (tensione positiva)
%MWr.m.c.24.9	DTR_OFF	Esplicito	BOOL	Reimposta il segnale DTR (tensione negativa)
%MWr.m.c.24.10		Esplicito	BOOL	Imposta il segnale RTS (tensione positiva)
%MWr.m.c.24.11		Esplicito	BOOL	Reimposta il segnale RTS (tensione negativa)
%MWr.m.c.24.12	TO_MODBUS_MASTER	Esplicito	BOOL	Commuta alla modalità master
%MWr.m.c.24.13	TO_MODBUS_SLAVE	Esplicito	BOOL	Commuta alla modalità slave
%MWr.m.c.24.14	TO_CHAR_MODE	Esplicito	BOOL	Passa alla modalità caratteri
%MWr.m.c.25	SLAVE_ADDR	Esplicito	INT	Indirizzo dello slave Modbus da memorizzare nella FLASH
%MWr.m.c.26		Esplicito	INT	LOW BYTE: conteggio tentativi master: numero di tentativi in modalità master [0..15] vedere %MW24.1 Inizio ritardo se %MW26.6 è impostato. Tempo di attesa dopo che il CTS è OK prima di iniziare ad inviare il frame. È utile per i modem che richiedono più tempo dopo il segnale CTS o che non gestiscono il segnale CTS (in questo caso il segnale RTS deve essere collegato al segnale CTS). Questo tempo è espresso in millisecondi, la precisione è di circa 3ms. Può essere eseguito solo in modalità RS232.
%MWr.m.c.27		Esplicito	INT	LOW BYTE: slave per il quale il master adatterà il ritardo della risposta [0..248, 255=ALL] vedere %MW24.2 e %MW28 Fine ritardo se %MW24.6 è impostato. Tempo di attesa dopo aver inviato un frame prima di rilasciare il segnale RTS, per lasciare al MODEM tempo sufficiente per completare l'invio del frame prima dell'interruzione del collegamento. Questo tempo è espresso in millisecondi, la precisione è di circa 3ms. Può essere eseguito solo in modalità RS232.
%MWr.m.c.28		Esplicito	INT	Ritardo di risposta specifico per uno slave in 10ms [1..1000] vedere %MW24.2 e %MW27
%MWr.m.c.29		Esplicito	INT	Tempo morto in 10ms [1..10] vedere %MW24.3

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.30		Esplicito	INT	T3,5char: ritardo tra i frame in millisecondi [0..10000]. Il valore utilizzato dipende dalla velocità. Se il valore è inferiore o superiore ai valori possibili, viene applicato il limite inferiore o il limite superiore e il comando viene accettato. Il valore 0 significa nessun cambiamento nel segnale RTU. Il ritardo della risposta viene ricalcolato.
%MWr.m.c.31		Esplicito	INT	T1,5char : ritardo tra i caratteri in millisecondi [0..9999]. Il valore utilizzato dipende dalla velocità. Se il valore è inferiore o superiore ai valori possibili, viene applicato il limite inferiore o il limite superiore e il comando viene accettato. Il valore 0 significa che T1,5 viene calcolato come T3,5ch – 2ch (calcolo predefinito).
%MWr.m.c.32		Esplicito	INT	Ritardo tra gli scambi master in modalità RTU [0..256] in millisecondi. Il valore 0 significa "nessun ritardo"; se il valore ha durata inferiore a 10 bit, viene usato il valore minimo 10 bit.

Esempio di codice

```
(* master sideNOM si trova nel rack 0 slot 9 *)
if HalfModemMaster then
 HalfModemMaster:=false;
 %MW0.9.0.24:=16#0450;(* passa alla modalità half duplex con RTS e cambia i tempi MODBUS*)
 %MW0.9.0.26:=12;(* 12ms di attesa prima dell'invio quando CTS aumenta *)
 %MW0.9.0.27:=9; (* lascia RTS attivo 9ms dopo la fine dell'invio *)
 %MW0.9.0.30:=0;
 %MW0.9.0.31:=0;(*usa il valore della schermata di configurazione uguale a 6ms *)
 %MW0.9.0.32:=50; (*50ms di ritardo prima di inviare un nuovo frame*)
 write_cmd(%ch0.9.0);(* invia il comando e i dati al canale NOM*)
end_if;
(* lato slave il NOM si trova nel rack 0 slot 3 *)
if HalfModemSlave then
 HalfModemSlave:=false;
 %MW0.3.0.24:=16#0448;(* passa alla modalità half duplex con RTS e cambia il tempo morto
slave*)
```

```
 %MW0.3.0.26:=12;(* 12ms di attesa prima dell'invio quando CTS aumenta *)
 %MW0.3.0.27:=9; (* lascia RTS attivo 9ms dopo la fine dell'invio *)
 %MW0.3.0.29:=4; (* 4*10ms di tempo morto *)
 write_cmd(%ch0.3.0);(* invia il comando e i dati al canale NOM*)
end_if;
(* opzionale: invio automatico del comando *)
if %S0 or %S1 or %S13 then
 memoSendCmd:=true;
end_if;
(* copia ad ogni ciclo l'errore del modulo per rilevare la scomparsa errore *)
memoSendCmd:=%I0.3.0.ERR;
(* se modulo OK invia il comando una volta *)
if FE(memoSendCmd) then
 HalfModemSlave:=true;
end_if;
```

NOM Internal Register Readable

I registri interni Nom sono accessibili solo in modalità MODBUS tramite l'EF READ_VAR. Esempio di codice (il modulo NOM si trova nel rack 0 slot 3):

```
if dataCh030GetChannelGlobalInfo then
  read_var(addm('0.3.0'), '%MW', 200, 3, dataCh030Mgt,  dataCh030Buff);
(* Internal_Reg@200 vengono copiati nel buffer dataCh030Buff *)
 dataCh030GetChannelGlobalInfo := false;
end_if;
```

- Internal_Reg@0 : inizio ritardo in ms (precisione circa 3ms) (accesso in lettura o scrittura)
- Internal_Reg@1 : fine ritardo in ms (precisione circa 3ms) (accesso in lettura o scrittura)
- Internal_Reg@200 : numero versione interfaccia = 1
- Internal_Reg@201 : indirizzo slave memorizzato nella FLASH
- Internal_Reg@202 : 1=possibile cambiare la FLASH, 0=vietato cambiarla
- Internal_Reg@1000 : codice interno RTU master Modbus ch0=1110, ch1=2110
- Internal_Reg@1002 : 0 = Full Duplex - Controllo flusso hardware o RS485 ; 1 = Half Duplex - Direzione gestita automaticamente dal modulo con RTS
- Internal_Reg@1010 : ritardo invio interno tra caratteri in bit (nbbits*1000/velocità => durata in ms) [T1,5S].
- Internal_Reg@1012 : ritardo ricezione interno tra caratteri in bit [T1,5R].
- Internal_Reg@1014 : ritardo invio interno tra frame in bit [T3,5S]
- Internal_Reg@1016 : ritardo ricezione interno tra frame in bit [T3,5R]

- Internal_Reg@1018 : tempo di attesa prima dell'invio dei frame successivi in bit.
- Internal_Reg@1090 : conteggio tentativi master.
- Internal_Reg@1100 : ritardo risposta slave per trasmissione broadcast in 10ms.
- Internal_Reg@1101 : ritardo risposta slave per slave 1 in 10ms.
- …
- Internal_Reg@1348 : ritardo risposta slave per indirizzamento punto-punto (248).
- Internal_Reg@1500 : codice interno slave RTU Modbus ch0=1120, ch1=2120
- Internal_Reg@1502 : 0 = Full Duplex - Controllo flusso hardware o RS485 ; 1 = Half Duplex - Direzione gestita automaticamente dal modulo con RTS
- Internal_Reg@1510 : ritardo invio interno tra caratteri in bit (nbbits*1000/velocità => durata in ms) [T1,5S].
- Internal_Reg@1512 : ritardo ricezione interno tra caratteri in bit [T1,5R].
- Internal_Reg@1514 : ritardo invio interno tra frame in bit [T3,5S].
- Internal_Reg@1516 : ritardo ricezione interno tra frame in bit [T3,5R].
- Internal_Reg@1518 : tempo di attesa prima dell'invio dei frame successivi in bit.
- Internal_Reg@1602 : tempo morto dopo ricezione in ms.
- Internal_Reg@1606 : modalità Solo ascolto attiva = 1, (non attiva = 0).
- Internal_Reg@2000 : codice interno ASCII master Modbus ch0=1210, ch1=2210
- Internal_Reg@2002 : 0 = Full Duplex - Controllo flusso hardware o RS485 ; 1 = Half Duplex - Direzione gestita automaticamente dal modulo con RTS
- Internal_Reg@2010 : ritardo invio interno tra caratteri in bit (nbbits*1000/velocità => durata in ms) [T1,5S].
- Internal_Reg@2012 : ritardo ricezione interno tra caratteri in bit [T1,5R].
- Internal_Reg@2014 : ritardo invio interno tra frame in bit [T3,5S].
- Internal_Reg@2014 : ritardo invio interno tra frame in bit [T3,5S].
- Internal_Reg@2014 : ritardo invio interno tra frame in bit [T3,5S].
- Internal_Reg@2016 : ritardo ricezione interno tra frame in bit [T3,5R].
- Internal_Reg@2018 : tempo di attesa prima dell'invio dei frame successivi in bit.
- Internal_Reg@2090 : conteggio tentativi master.
- Internal_Reg@2100 : ritardo risposta slave per trasmissione broadcast in 10ms.
- Internal_Reg@2101 : ritardo risposta slave per slave 1 in 10ms.
- …
- Internal_Reg@2348 : ritardo risposta slave per indirizzamento punto-punto (248).
- Internal_Reg@2500 : codice interno ASCII slave Modbus ch0=1220, ch1=2220
- Internal_Reg@2502 : 0 = Full Duplex - Controllo flusso hardware o RS485 ; 1 = Half Duplex - Direzione gestita automaticamente dal modulo con RTS
- Internal_Reg@2510 : ritardo interno tra caratteri in bit durante l'invio (nbbits*1000/velocità => durata in ms) [T1,5S].
- Internal_Reg@2512 : ritardo interno tra caratteri in bit durante la ricezione [T1,5R].
- Internal_Reg@2514 : ritardo interno tra frame in bit durante l'invio [T3,5S].
- Internal_Reg@2516 : ritardo interno tra frame in bit durante la ricezione [T3,5R].
- Internal_Reg@2518 : tempo di attesa in bit prima dell'invio del frame successivo.
- Internal_Reg@2600 : indirizzo slave in uso.
- Internal_Reg@2602 : tempo morto dopo ricezione in ms.
- Internal_Reg@2606 : modalità Solo ascolto attiva = 1, (non attiva = 0).

- Internal_Reg@3000 : codice interno modalità caratteri ch0=1000, ch1=2000
- Internal_Reg@3002 : 0 = Full Duplex - Controllo flusso hardware o RS485 ; 1 = Half Duplex - Direzione gestita automaticamente dal modulo con RTS
- Internal_Reg@3100 : 0=nessun criterio di stop attivo, 1 stop su silenzio o stop a fine caratteri
- Internal_Reg@3102 : silenzio interno in bit (min 2 bit, max 65535 bit)
- Internal_Reg@3104 : primo byte di fine frame da utilizzare, 16#0100 significa nessun byte
- Internal_Reg@3106 : primo EOF: 1=byte di fine frame per accettare il frame, 0=rimuove il byte di fine frame
- Internal_Reg@3108 : secondo byte di fine frame
- Internal_Reg@3110 : secondo EOF: 1=byte di fine frame per accettare il frame, 0=rimuove il byte di fine frame

Sezione 7.4
Debug della comunicazione Modbus Seriale

Schermata di debug della comunicazione Modbus

Informazioni generali

La schermata di debug della comunicazione Modbus seriale è accessibile solo in modalità online.

Accesso alla schermata di debug

La tabella seguente descrive la procedura per accedere alla schermata di debug della comunicazione Modbus seriale:

Passo	Azione
1	Accedere alla schermata di configurazione della comunicazione Modbus seriale. *(vedi pagina 134)*
2	Selezionare la scheda "Debug" sulla schermata visualizzata.

Descrizione della schermata Debug

La schermata di debug è suddivisa in due o tre zone:
- La zona Tipo e la zona numero slave,
- La zona Contatori,
- La zona Segnali (se RS232).

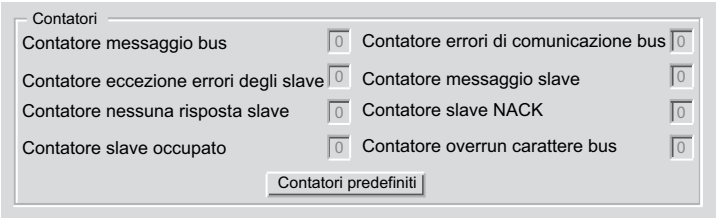
La zona Tipo e la zona numero slave

Se il modulo ha la funzione di Master nel collegamento Modbus, questa zona è come segue:

Se il modulo ha la funzione di Slave nel collegamento Modbus, questa zona è come segue:

La zona Contatori

La zona è illustrata di seguito:



Questa zona mostra i vari contatori di debug.

Il pulsante Azzera contatori azzera tutti i contatori della modalità di debug.

Funzionamento dei contatori

I contatori di debug della comunicazione Modbus seriale sono i seguenti:

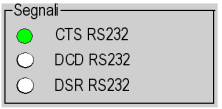
- **Contatore messaggio bus**: questo contatore indica il numero di messaggi individuati dal modulo sul collegamento seriale. I messaggi con un risultato di controllo CRC negativo non vengono contati.
- **Contatore errori di comunicazione bus**: questo contatore indica il numero di risultati di controllo CRC negativi contati dal modulo. Se viene rilevato un errore di carattere (overflow, errore di parità), oppure se il messaggio è più corto di 3 byte, il sistema che riceve i dati non può eseguire il controllo CRC. In tali casi, il contatore viene incrementato di conseguenza.
- **Contatore di errore eccezione slave**: questo contatore indica il numero di errori di eccezione Modbus individuati dal modulo.
- **Contatore messaggio slave**: questo contatore indica il numero di messaggi ricevuti ed elaborati dal collegamento Modbus.
- **Contatore "nessuna risposta" slave**: questo contatore indica il numero di messaggi inviati dal sistema remoto per cui non ha ricevuto risposta (normale o di eccezione). Conteggia, inoltre, il numero di messaggi ricevuti in modalità trasmissione.
- **Contatore riconoscimento slave negativo**: questo contatore indica il numero di messaggi inviati al sistema remoto per cui è stato restituito un riconoscimento negativo.
- **Contatore slave occupato**: questo contatore indica il numero di messaggi inviati al sistema remoto per cui è stato restituito un messaggio di eccezione "slave occupato".
- **Contatore overflow carattere bus**: questo contatore indica il numero di messaggi inviati al modulo che non è stato possibile acquisire a causa di un overflow di caratteri sul bus. L'overflow è causato da:
  - dati di tipo carattere che vengono trasmessi sulla porta seriale più rapidamente di quanto non sia possibile memorizzarli
  - una perdita di dati dovuta all'hardware.

NOTA: per tutti i contatori, il conteggio inizia dall'avvio più recente, dall'operazione di azzeramento contatori o dall'accensione del modulo.

La zona Segnali

Questa zona viene visualizzata solamente se RS232 viene selezionato nella schermata di configurazione. Se RS485 viene selezionato nella schermata di configurazione, questa finestra non viene visualizzata per nulla.

La zona Segnali è illustrata di seguito:



Questa zona indica l'attività dei segnali:

- **CTS RS232**: indica l'attività del segnale CTS.
- **DCD RS232**: indica l'attività del segnale DCD.
- **DSR RS232**: indica l'attività del segnale DSR.

Capitolo 8
Comunicazione Modalità carattere per BMX NOM 0200

Oggetto della sezione

Questo capitolo illustra il processo di implementazione software per la comunicazione con Modalità carattere per BMX NOM 0200.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
8.1	Generalità	170
8.2	Configurazione comunicazione in modalità carattere	171
8.3	Programmazione comunicazione in modalità carattere	187
8.4	Debug della comunicazione in modalità carattere	200

Sezione 8.1
Generalità

Informazioni sulla comunicazione in modalità carattere

Introduzione

La comunicazione in modalità carattere consente l'esecuzione di funzioni di comunicazione e dialogo tra i PLC e i dispositivi seguenti:

- periferiche normali (stampanti, schermi tattili, terminali di laboratorio, ecc.),
- periferiche specializzate (lettori di codice a barre, ecc.),
- calcolatori (controllo, gestione produzione, ecc.),
- dispositivi diversi (comandi numerici, controllori a velocità variabile, ecc.),
- Modem esterno.

⚠ AVVERTENZA
PERDITA CRITICA DEI DATI Le porte di comunicazione devono essere usate solo per il trasferimento di dati non critici. **Il mancato rispetto di queste istruzioni può provocare morte, gravi infortuni o danni alle apparecchiature.**

Sezione 8.2
Configurazione comunicazione in modalità carattere

Oggetto della sezione

Questa sezione descrive il processo di configurazione utilizzato per implementare la comunicazione in modalità carattere.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in un rack locale	172
Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in derivazione X80	175
Funzioni accessibili in modalità carattere	178
Valori predefiniti dei parametri di comunicazione in modalità carattere	179
Parametri di rilevamento fine messaggio in modalità carattere	180
Parametri di trasmissione in modalità carattere	182
Parametri del segnale e della linea fisica in modalità carattere	184

Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in un rack locale

Informazioni generali

Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modalità Carattere.

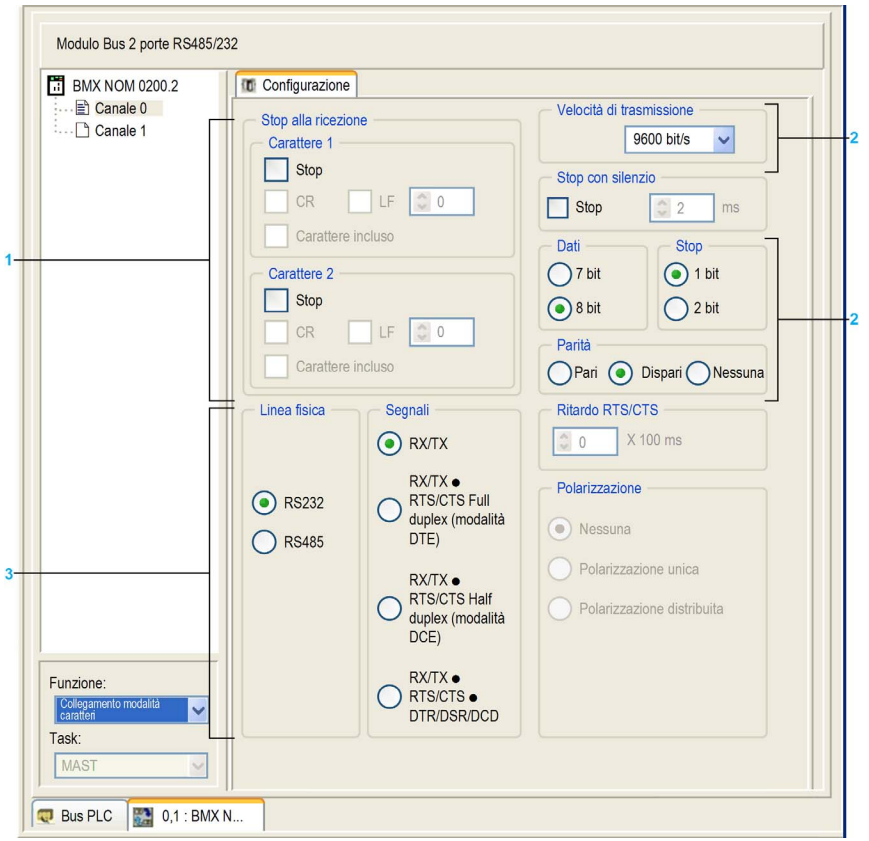
Accesso alla schermata di configurazione

La tabella seguente descrive la procedura per accedere alla schermata di configurazione della comunicazione in modalità carattere.

Passo	Azione
1	Aprire la sottodirectory BMX NOM 0200 nel browser di progetto *(vedi pagina 130)*.
2	Selezionare il Canale da configurare e la funzione collegamento modalità carattere sullo schermo che compare.

Schermata di configurazione in modalità carattere

La figura di seguito mostra la schermata di configurazione predefinita per la comunicazione Modalità carattere sul Canale 0:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione della Modalità Carattere:

Legenda	Elemento	Commento
1	Parametri di individuazione fine messaggio *(vedi pagina 180)*	Tali parametri sono accessibili da due zone: ● Stop alla ricezione, ● Stop con silenzio.
2	Parametri di trasmissione *(vedi pagina 182)*	Tali parametri sono accessibili da quattro zone: ● Velocità di trasmissione, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 184)*	Tali parametri sono accessibili da quattro zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS, ● Polarizzazione,

NOTA: In questo esempio, le zone "Polarizzazione" e "Ritardo RTS/CTS" sono disattivate perché è stata selezionata una linea fisica RS232 e sono stati selezionati i segnali RX/TX.

Schermata di configurazione per la comunicazione in modalità carattere del modulo BMX NOM 0200 in derivazione X80

Informazioni generali

Le pagine seguenti costituiscono un'introduzione alla schermata di configurazione per la comunicazione Modalità Carattere.

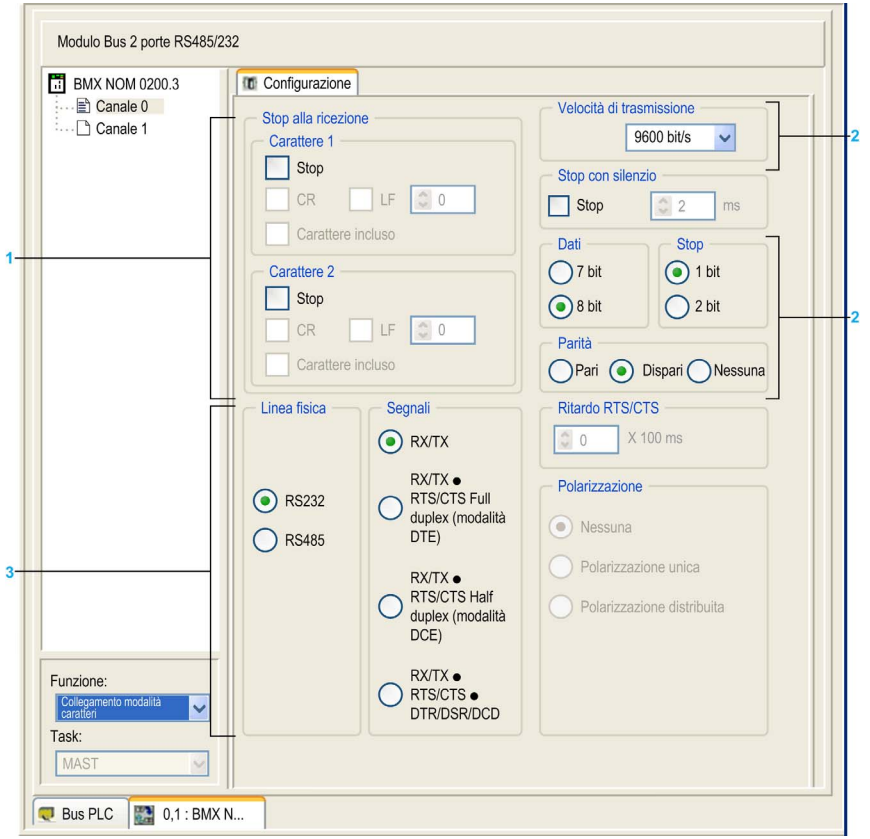
Accesso alla schermata di configurazione

La tabella seguente descrive la procedura per accedere alla schermata di configurazione della comunicazione in modalità carattere.

Passo	Azione
1	Aprire la sottodirectory BMX NOM 0200 nel browser di progetto *(vedi pagina 130)*.
2	Selezionare il Canale da configurare e la funzione collegamento modalità carattere sullo schermo che compare.

Schermata di configurazione in modalità carattere

La figura di seguito mostra la schermata di configurazione predefinita per la comunicazione Modalità carattere sul Canale 0:



Descrizione

Queste zone vengono usate per configurare i parametri del canale; nella modalità online, queste zone sono accessibili. Nella modalità offline, queste zone sono accessibili, ma alcuni parametri potrebbero non essere accessibili e quindi essere disabilitati.

La seguente tabella mostra le diverse zone della schermata di configurazione della Modalità Carattere:

Legenda	Elemento	Commento
1	Parametri di individuazione fine messaggio *(vedi pagina 180)*	Tali parametri sono accessibili da due zone: ● Stop alla ricezione, ● Stop con silenzio.
2	Parametri di trasmissione *(vedi pagina 182)*	Tali parametri sono accessibili da quattro zone: ● Velocità di trasmissione, ● Dati, ● Bit di stop, ● Parità.
3	Parametri di linea fisica e segnale *(vedi pagina 184)*	Tali parametri sono accessibili da quattro zone: ● Linea fisica, ● Segnali, ● Ritardo RTS/CTS, ● Polarizzazione,

NOTA: In questo esempio, le zone "Polarizzazione" e "Ritardo RTS/CTS" sono disattivate perché è stata selezionata una linea fisica RS232 e sono stati selezionati i segnali RX/TX.

Funzioni accessibili in modalità carattere

In breve

L'accessibilità delle funzioni per la configurazione del collegamento seriale di un modulo BMX NOM 0200 tramite il protocollo Modalità carattere dipende dal collegamento fisico utilizzato.

Funzioni accessibili

La tabella seguente mostra le varie funzioni configurabili in base al tipo di collegamento seriale utilizzato.

Funzione	Collegamento RS 485 (Canale 0 o Canale 1)	Collegamento RS 232 (Canale 0)
Velocità di trasmissione	X	X
Dati	• 7 bit • 8 bit	• 7 bit • 8 bit
Stop	• 1 bit • 2 bit	• 1 bit • 2 bit
Parità	• Odd • Pari • Nessuno	• Odd • Pari • Nessuno
Stop alla ricezione	X	X
Stop con silenzio	X	X
Segnali RX/TX	X	X
Segnali RTS/CTS	-	X
Ritardo RTS/CTS	-	X
Segnali DTR/DSR/DCD	-	X
Polarizzazione	X	-

X Funzione accessibile
- Funzione inaccessibile

Valori predefiniti dei parametri di comunicazione in modalità carattere

In breve

Tutti i parametri di comunicazione in modalità carattere presentano valori predefiniti.

Valori predefiniti

La tabella di seguito mostra i valori predefiniti per i parametri di comunicazione Modalità carattere sul Canale 0 e il Canale 1 del modulo BMX NOM 0200:

Parametro di configurazione	Valore sul Canale 0	Valore sul Canale 1
Linea fisica	RS232	RS485
Segnali	RX/TX	RX/TX (valore unico)
Velocità di trasmissione	9600 bit/s	9600 bit/s
Parità	Odd	Odd
Bit di dati	8 bit	8 bit
Bit di stop	1 bit	1 bit
Polarizzazione	Nessuno (valore unico)	Nessuno

Parametri di rilevamento fine messaggio in modalità carattere

In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di individuazione fine messaggio.

Tali parametri sono accessibili da due zone:

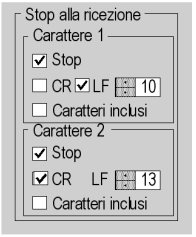
- La zona Stop alla ricezione: si arresta alla ricezione di un carattere speciale;
- La zona Stop con silenzio: si arresta in presenza di silenzio.

Condizioni di utilizzo

Se si seleziona Stop con silenzio, si deseleziona lo Stop alla ricezione e viceversa.

La zona Stop alla ricezione

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



È possibile terminare una richiesta di ricezione dopo la ricezione di un carattere specifico.

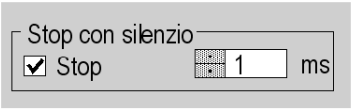
Selezionando l'opzione Stop, è possibile configurare Stop alla ricezione in modo da attivarla da uno specifico carattere di fine messaggio:

- CR: consente di individuare la fine del messaggio tramite un ritorno a capo;
- LF: consente di individuare la fine del messaggio tramite un avanzamento di riga;
- Campo inserimento dati: consente di identificare un carattere di fine messaggio diverso dai caratteri CR o LF, mediante un valore decimale:
  - tra 0 e 255 se i dati sono codificati a 8 bit
  - tra 0 e 127 se i dati sono codificati a 7 bit
- Carattere incluso: consente di includere un carattere di fine messaggio nella tabella di ricezione dell'applicazione del PLC.

È possibile configurare due caratteri di fine ricezione. Nella finestra seguente, il termine della ricezione di un messaggio è individuato da un carattere LF o CR.

La zona Stop con silenzio

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



Questa zona consente di individuare la fine di un messaggio in ricezione tramite l'assenza dei caratteri di fine messaggio in un intervallo determinato.

Stop con silenzio viene convalidata selezionando la casella Stop. La durata del silenzio (espressa in millisecondi) viene impostata tramite il campo di inserimento dati.

NOTA: i valori disponibili sono compresi tra 1ms e 10000ms e dipendono dalla velocità di trasmissione selezionata.

Parametri di trasmissione in modalità carattere

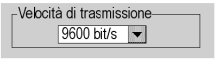
In breve

Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di trasmissione.

Tali parametri sono accessibili da quattro zone:
- La zona Velocità di trasmissione
- La zona Dati,
- La zona Stop,
- La zona Parità.

La zona Velocità di trasmissione

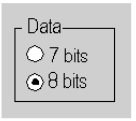
Questa zona di configurazione appare sullo schermo come illustrato di seguito:



È possibile utilizzarla per selezionare la velocità di trasmissione del protocollo Modalità carattere. La velocità selezionata deve essere coerente con gli altri dispositivi. I valori configurabili sono 300, 600, 1200, 2400, 4800, 9600, 19200, 57600 e 115200 (solo sul canale 0 in modalità RS232) bit al secondo.

La zona Dati

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



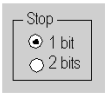
In questa zona è possibile specificare la dimensione dei dati da scambiare sul collegamento.

I valori disponibili sono:
- 7 bit
- 8 bit

All'utente viene richiesto di regolare il numero di bit di dati in base al dispositivo remoto utilizzato.

La zona Stop

La zona è illustrata di seguito:



La zona Stop consente di inserire il numero di bit di stop utilizzati per la comunicazione. All'utente viene richiesto di regolare il numero di bit di stop in base al dispositivo remoto utilizzato.

I valori configurabili sono:
- 1 bit
- 2 bit

La zona Parità

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

Questa zona consente di stabilire l'aggiunta o meno di un bit di parità e di definirne il tipo. All'utente viene richiesto di regolare la parità in base al dispositivo remoto utilizzato.

I valori configurabili sono:
- Pari
- Odd
- Nessuno

Parametri del segnale e della linea fisica in modalità carattere

In breve

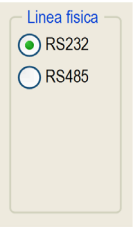
Dopo aver configurato il canale di comunicazione, è necessario specificare i parametri di segnale e linea fisica.

Tali parametri sono accessibili da tre zone:

- La zona Linea fisica
- La zona Segnali
- La zona Ritardo RTS/CTS

La zona Linea fisica

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

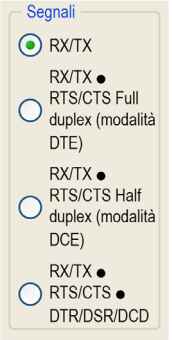


In questa zona, è possibile scegliere due tipi di linea fisica per la porta seriale sul modulo BMX NOM 0200:

- La linea RS 232
- La linea RS 485

La zona Segnali

Questa zona di configurazione appare sullo schermo come illustrato di seguito:



In questa zona, è possibile selezionare i segnali supportati dalla linea fisica RS 232:

- RX/TX
- RX/TX + RTS/CTS Full Duplex (modalità DTE)
- RX/TX + RTS/CTS Half Duplex (modalità DCE)
- RX/TX + RTS/CTS + DTR/DSR/DCD

Se è configurata la linea RS 485, tutta la zona sarà disattivata e il valore predefinito sarà RX/TX.

La RTS/CTS zona Ritardo

Questa zona di configurazione appare sullo schermo come illustrato di seguito:

La zona ritardo RTS/CTS è disponibile solamente quando entrambe le caselle di selezione RS232 e RX/TX+RTS/CTS o RX/TX+RTS/CTS+DTR/DSR/DCD sono selezionate. Viene eseguito un controllo del flusso hardware RTS/CTS.

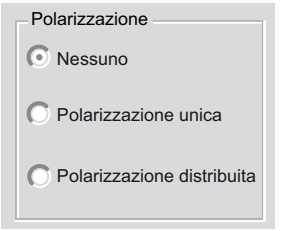
L'algoritmo di controllo del flusso hardware RTS/CTS ha lo scopo di prevenire il buffer di ricezione dell'overflow (full duplex).

Il RTS/CTS ritardo corrisponde al ritardo del time out tra l'innalzamento RTS e l'innalzamento CTS. Un valore di ritardo RTS/CTS diverso da 0 inoltre corrisponde al tempo massimo di attesa tra ciascuna trasmissione di caratteri dopo l'innalzamento dei segnali RTS e CTS. Se il valore è impostato sullo 0, gli UART possono rimanere bloccati in uno stato di attesa per un tempo infinito finché il CTS si innalza, e pertanto il valore 0 viene utilizzato solamente in casi particolari come per collegare ad anello il segnale RTS con il segnale CTS e questo per verificare che nessun cavo di collegamento sia difettoso.

NOTA: L'impostazione predefinita è 0 ms.

La zona Polarizzazione

La zona di configurazione illustrata di seguito è accessibile solo se si seleziona "RS485" nella zona "Linea fisica":



Questa zona dà la possibilità di scegliere tra tre tipi di configurazione per la polarizzazione sul canale:

- **Nessuna** per non usare nessuna polarizzazione nel caso in cui si possieda la propria terminazione.
- **Polarizzazione unica** per usare una polarizzazione a bassa impedenza come nelle reti Modbus (lo scopo di questo tipo di polarizzazione è di consentire al master di mantenere lo stato predefinito).
- **Polarizzazione distribuita** per usare una polarizzazione ad alta impedenza (lo scopo di questo tipo di polarizzazione è di consentire a ciascun dispositivo di mantenere lo stato predefinito).

Sezione 8.3
Programmazione comunicazione in modalità carattere

Oggetto della sezione

Questa sezione descrive il processo di programmazione utilizzato per implementare la comunicazione in modalità caratteri.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Funzioni di comunicazione in Modalità caratteri	188
Dettagli della modalità Expert in modalità Carattere	197

Funzioni di comunicazione in Modalità caratteri

Funzioni disponibili

Per inviare e ricevere i dati tramite un canale di comunicazione in modalità caratteri sono definite tre funzioni di comunicazione specifiche:

- `PRINT_CHAR` invia una stringa di caratteri con un massimo di 16 x 1.024 byte.
- `INPUT_CHAR` legge una stringa di caratteri con un massimo di 16 x 1.024 byte.
- `INPUT_BYTE` legge array di byte con un massimo di 16 x 1.024 byte.

Il BMX NOM 0200 è in grado di memorizzare un totale di 16 frame in trasmissione o ricezione. I frame nei buffer sono gestiti in ordine FIFO. Sulle linee RS-232 sono invece gestiti in modalità full duplex.

NOTA: Per la funzione `INPUT_CHAR`, quando il canale è configurato senza **Stop con silenzio** è necessario configurare un timeout affinché avvenga il riconoscimento del bit di attività della funzione.

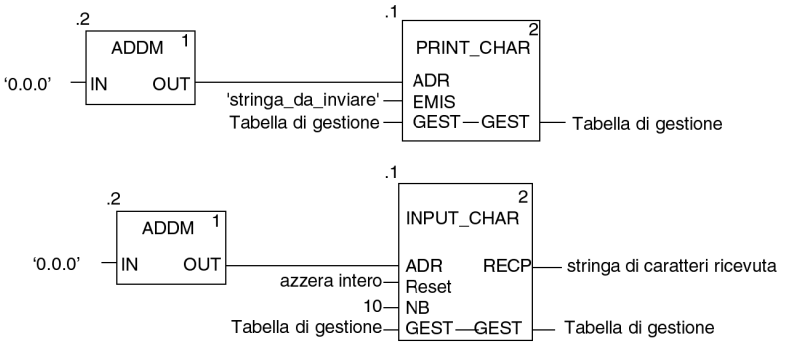
Numero di frame ricevuti nei buffer

Quando la porta Modbus è configurata in modalità caratteri, %MWr.m.c.7 indica il numero di frame nel buffer di ricezione del BMX NOM 0200.

Questa PAROLA viene incrementata ogni volta che il BMX NOM 0200 riceve un frame sulla linea RS-232.

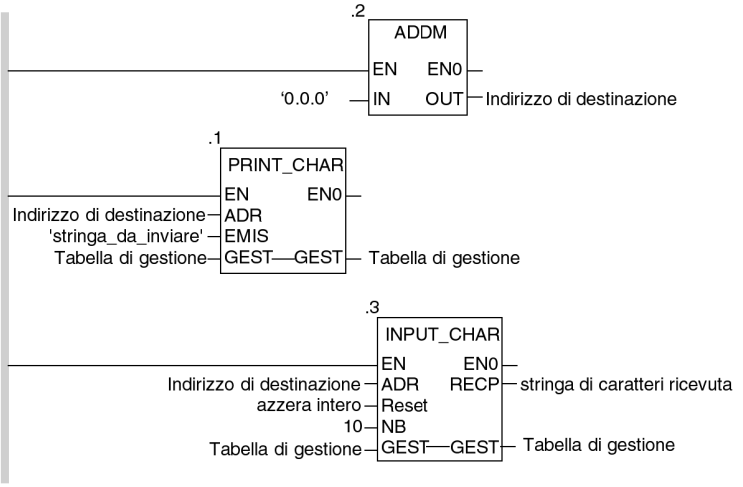
Esempio di programmazione in FBD

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione `PRINT_CHAR` e `INPUT_CHAR` nel linguaggio FBD:



Esempio di programmazione in Ladder

Lo schema seguente mostra un esempio di programmazione delle funzioni di comunicazione `PRINT_CHAR` e `INPUT_CHAR` nel linguaggio Ladder:



Esempio di programmazione in ST

Le righe di codice seguenti mostrano un esempio di programmazione delle funzioni di comunicazione `PRINT_CHAR` e `INPUT_CHAR` nel linguaggio ST:

```
PRINT_CHAR(ADDM('0.1.0'), 'stringa da inviare, Gestione tabella);
INPUT_CHAR(ADDM('0.1.0'), reimposta a 0, 10, Tabella di gestione, stringa
di caratteri ricevuta);
```

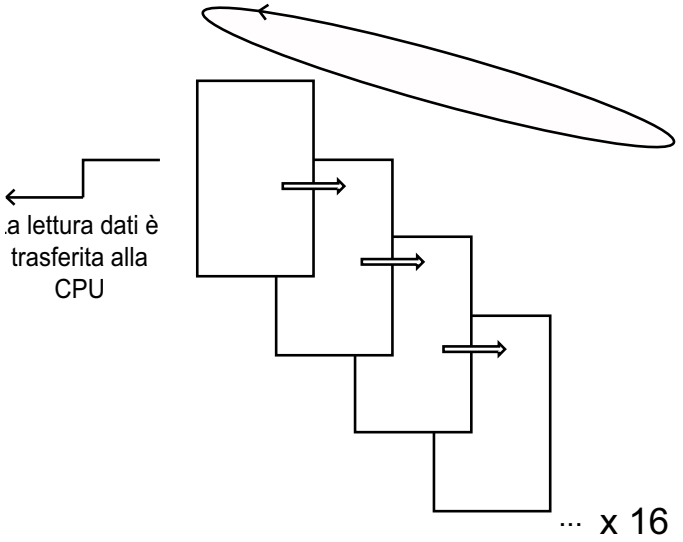
Caratteristica della funzione INPUT_CHAR

Se il parametro di ingresso Reset è impostato a 1, tutti i buffer vengono prima azzerati, quindi il modulo attende la ricezione dei dati. Si consiglia di utilizzare questa funzione per iniziare correttamente la ricezione rimuovendo i vecchi dati dai buffer.

Meccanismo interno del modulo BMX NOM 0200

I dati ricevuti vengono archiviati in 16 buffer ciclici in serie, ognuno dei quali contiene 1024 byte.

La seguente figura illustra questo meccanismo:



Ricezione di dati con INPUT_CHARO INPUT_BYTE

I frame vengono recuperati dal programma applicativo utilizzando l'EF INPUT_CHAR per ricevere una stringa, oppure l'EF INPUT_BYTE per ricevere dati binari.

L'EF INPUT_CHAR o INPUT_BYTE può essere eseguito prima che il modulo riceva i dati. In tal caso, il modulo attende i dati dalla linea, quindi li invia alla CPU.

L'EF può inoltre essere eseguito quando il frame è già stato ricevuto (ad esempio, dopo il controllo di %MWr.m.c.7 con READ_STS). In tal caso, il modulo invia immediatamente il frame con buffer alla CPU.

E' anche possibile forzare il modulo in modo che attenda i dati dalla linea impostando il parametro Reset dell'EF a 1. In questo caso, i dati precedentemente immessi nel buffer vengono eliminati e il BMX NOM 0200 attende nuovi dati da inviare alla CPU.

Modalità di ricezione

Il modulo BMX NOM 0200 è in grado di memorizzare un massimo di 16 frame.

Il BMX NOM 0200 può essere impostato internamente nelle modalità messaggio o grezza indipendentemente dalla definizione o meno dei criteri di fine frame:

- La modalità messaggio viene impostata se è stata specificata una condizione di fine frame. In questa modalità, il parametro NB di `INPUT_CHAR` o `INPUT_BYTE` è la quantità di byte inviati alla CPU.
  I valori speciali NB = 0 e NB = 1024 indicano che l'intero frame è stato inviato alla CPU.
  Gli altri valori di NB indicano il numero di byte da inviare alla CPU. Se prima della chiamata di `INPUT_CHAR` o `INPUT_BYTE` sono stati ricevuti altri byte, quelli rimanenti verranno ignorati.
- La modalità grezza viene impostata se non è stata specificata una condizione di fine frame. La Modalità caratteri non è un protocollo "orientato ai frame". In ricezione, una fine frame viene rilavata dal primo di questi eventi:
  - una durata di silenzio specificata nella zona **Stop con silenzio** della schermata di configurazione della comunicazione in modalità carattere *(vedi pagina 172)*
  - è stato ricevuto il carattere 1 o 2, come specificato nella zona **Stop alla ricezione** della schermata di configurazione della comunicazione in modalità carattere *(vedi pagina 172)*
  - sono stati ricevuti più di 1024 byte

Se viene rilevata una fine frame, il contatore interno %MWr.m.c.7 viene incrementato.

Dimensioni massime dei dati

Le dimensioni massime di un frame inviato dal BMX NOM 0200 alla CPU sono pari a 1024 byte. Tuttavia, internamente, le dimensioni massime del frame di ricezione sono di 1025 byte se viene configurato un byte di fine frame e tale byte non deve essere incluso nei dati inviati alla CPU.

Frame di dimensioni zero

I frame di dimensioni zero vengono ignorati. Se viene configurato un byte di fine frame (non richiesto come parte dei dati), i frame di dimensioni zero ricevuti dal BMX NOM 0200 non verranno inviati alla CPU. In tal caso, se viene ricevuto un byte di fine frame senza prima alcun dato, il frame verrà ignorato e alla CPU non verranno inviate informazioni.

Ricezione di più frame durante un task MAST

Durante un task MAST, il BMX NOM 0200 può inoltrare più frame alla CPU e in parallelo è possibile avviare svariate istanze EF INPUT_CHAR indirizzate allo stesso modulo BMX NOM 0200. Questa operazione può essere necessaria se sulla linea seriale arriva una quantità elevata di dati.

Annulla e Timeout

Annulla e Timeout vengono inoltrati al modulo BMX NOM 0200. La condizione Timeout e gli ordini Annulla applicati ad un'istanza di `INPUT_CHAR` vengono inoltrati al modulo BMX NOM 0200. Il task in sospeso corrispondente viene rimosso dalla coda dei task del modulo BMX NOM 0200.

Meccanismo interno del modulo BMX NOM 0200: emissione

Utilizzare l'EF PRINT_CHAR per inviare i dati su una linea seriale del modulo BMX NOM 0200.

NOTA: Se sono stati inviati più frame (sono state chiamate numerose istanze di PRINT_CHAR) ed è stato configurato un silenzio, tra ogni frame il modulo BMX NOM 0200 inserirà un tempo di silenzio.

E' possibile avviare fino a 16 richieste PRINT_CHAR: queste vengono inviate in modo seriale con un silenzio tra ogni PRINT_CHAR.

Annullamento di uno scambio

Esistono due modi di programmazione che consentono di annullare uno scambio eseguito dalle funzioni di comunicazione PRINT_CHAR e INPUT_CHAR. Essi sono presentati di seguito in linguaggio ST:

- Uso della funzione CANCEL:

```
IF (%MW40.0) THEN
 %MW200:=SHR(%MW40,8;)
 CANCEL(%MW200,%MW185);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione PRINT_CHAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma esegue le istruzioni seguenti:
  - Sposta i bit %MW40 di un byte (8 bit) a destra e carica il byte corrispondente al numero di scambio della comunicazione nella parola %MW200 .
  - Annulla lo scambio il cui numero è contenuto nella parola %MW200 mediante la funzione CANCEL.

- Tramite il bit di annullamento della funzione di comunicazione:

```
IF (%MW40.0) THEN
 SET(%MW40.1);
 PRINT_CHAR(ADDM('0.1.0'), 'stringa da inviare, %MW40:4);
END_IF;
```

  %MW40 è il parametro GEST (tabella gestione). %MW40.0 corrisponde al bit di attività della funzione PRINT_CHAR ed è impostato a 1 quando la funzione di comunicazione è attiva. Se questo bit è impostato a 1, il programma imposta il bit %MW40.1, il bit di annullamento della funzione, a 1. Quest'azione interrompe la comunicazione della funzione PRINT_CHAR.

NOTA: Quando si utilizza il bit di annullamento della funzione di comunicazione, occorre richiamare la funzione per attivare il bit di annullamento contenuto nella parola di gestione di scambio della funzione (%MW40 in questo esempio).

NOTA: quando si utilizza il bit di annullamento della funzione di comunicazione, è possibile annullare una comunicazione da una tabella di animazione impostando semplicemente il bit di annullamento della funzione a 1 (in questo esempio %MW40.1).

NOTA: la funzione CANCEL utilizza una parola di rapporto (`%MW185` in questo esempio).

NOTA: Questo esempio di programmazione è relativo alla funzione `PRINT_CHAR`, ma è applicabile anche alla funzione `INPUT_CHAR`.

Descrizione dei parametri della funzione ADDM

La tabella seguente descrive i diversi parametri della funzione `ADDM`:

Parametro	Tipo	Descrizione
IN	STRING	Indirizzo del dispositivo su bus o collegamento seriale. La sintassi dell'indirizzo è del tipo 'r.m.c.node'. L'indirizzo è composto dai seguenti parametri: ● **r**: numero di rack del sistema di destinazione, (sempre = 0). ● **m**: numero di slot del sistema di destinazione all'interno del rack, sempre = 0. ● **c**: numero di canale, sempre = 0 in quanto il collegamento seriale di un sistema remoto è sempre il canale 0. ● **nodo**: campo opzionale che può essere `SYS` o vuoto.
OUT	ARRAY [0..7] OF INT	Tabella con l'indirizzo di un dispositivo. Questo parametro può essere usato come un parametro d'ingresso per diverse funzioni di comunicazione.

Descrizione dei parametri della funzione PRINT_CHAR

La tabella seguente descrive i diversi parametri della funzione `PRINT_CHAR`:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo del canale della modalità caratteri di ricezione del messaggio dato dal parametro OUT della funzione ADDM.
EMIS	STRING	Stringa di caratteri da inviare.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola di rango 1: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: numero di scambio
		  ● Byte meno significativo: bit di attività (rango 0) e bit di annullamento (rango 1)
		● Parola di rango 2: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: rapporto operazione
		  ● Byte meno significativo: rapporto comunicazione
		● Parola di rango 3: una parola gestita dall'utente che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola di rango 4: una parola gestita dall'utente che definisce la lunghezza dello scambio:
		  ● Se la lunghezza di questo parametro è impostata a 0, il sistema invierà la stringa completa.
		  ● Se la lunghezza di questo parametro è maggiore della lunghezza della stringa, allora verrà restituito l'errore 16#0A (Dimensioni del buffer di emissione non sufficienti) nella seconda parola di gestione e non verrà inviato alcun carattere.

Descrizione dei parametri della funzione INPUT_CHAR

La tabella seguente descrive i diversi parametri della funzione `INPUT_CHAR`:

Parametro	Tipo	Descrizione
ADR	ARRAY [0..7] OF INT	Indirizzo del canale della modalità caratteri di ricezione del messaggio dato dal parametro OUT della funzione ADDM.
Reset	INT	Questo parametro può assumere due valori: ● Valore 1: reimposta memoria ricezione modulo a 0 **NOTA:** %MW.r.m.7 è reimpostato a 0. ● Valore 0: non reimpostare memoria ricezione modulo a 0
NB	INT	Lunghezza della stringa di caratteri da estrarre.

GEST	ARRAY [0..3] OF INT	Tabella di gestione scambio contenente le parole seguenti:
		● Parola di rango 1: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: numero di scambio
		  ● Byte meno significativo: bit di attività (rango 0) e bit di annullamento (rango 1)
		● Parola di rango 2: una parola gestita dal sistema e comprendente due byte:
		  ● Byte più significativo: rapporto operazione
		  ● Byte meno significativo: rapporto comunicazione
		● Parola di rango 3: una parola gestita dall'utente che definisce il tempo di risposta massimo mediante una base tempo di 100 ms.
		● Parola di rango 4: una parola gestita dall'utente che definisce la lunghezza dello scambio.

RECP	STRING	Stringa di caratteri ricevuta. Viene memorizzata in una stringa di caratteri.

Descrizione dei parametri della funzione INPUT_BYTE

La tabella seguente descrive i diversi parametri della funzione INPUT_BYTE:

Parametro	Tipo	Commento
ADR	ARRAY [0..7] OF INT	Per PLC Modicon M340: ● L'indirizzo del canale in modalità carattere di ricezione del messaggio è fornito dalla funzione ADDM. ● La sintassi dell'indirizzo è del tipo ADDM ('r.m.c.node'). Il nodo è un campo opzionale che può essere SYS o vuoto (ad es. ADDM('0.0.0.SYS') corrisponde a ADDM('0.0.0').
RAZ	INT	Reimpostazione. Questo parametro viene utilizzato per reimpostare la memoria di ricezione dell'accoppiatore. ● valore = 0: nessun reset della memoria ● valore = 1: reset della memoria **NOTA:** %MW.r.m.7 è reimpostato a 0. **NOTE**: sui PLC Modicon M340, l'EF INPUT_BYTE può essere programmato con o senza questo parametro.
NB	INT	Lunghezza del buffer o numero di byte da ricevere. ● Valore = 0: lettura del messaggio non appena è disponibile sul canale. In questo caso, nella schermata di configurazione occorre specificare una condizione di arresto. ● Valore maggiore di 0: specifica il numero di byte da leggere. **NOTA**: il carattere di fine messaggio predefinito è un ritorno a capo (CR).

Durata minima del silenzio

Se la casella di controllo "Stop" è selezionata, un silenzio sulla linea di ingresso è una delle condizioni che determinano il rilevamento di una fine frame.

Il valore minimo di questa durata equivale al tempo corrispondente della trasmissione di 1,5 caratteri. Espressa in numero di bit, e in base alla configurazione dei bit di avvio e di stop, la durata minima del silenzio è come segue:

Lunghezza totale di caratteri (bit)	Durata minima del silenzio (bit)
8	12
9	12
10	15
11	15

Convertire il numero nella colonna di destra in tempo, in base alla velocità di trasmissione configurata.

Dettagli della modalità Expert in modalità Carattere

Comunicazione in modalità Expert

La modalità Expert è un insieme di comandi che possono essere impartiti dal modulo per ottenere funzionalità aggiuntive.

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.24	CONTROL	Esplicito	INT	Segnale di comando, cambia protocollo.
%MWr.m.c.24.0		Esplicito	BOOL	Cancella i contatori locali.
%MWr.m.c.24.4		Esplicito	BOOL	Modificare la temporizzazione interna del silenziamento (%MW30). L'aggiornamento di questo valore può creare interferenze se il modulo è in funzione.
%MWr.m.c.24.5		Esplicito	BOOL	Modificare la modalità carattere fine frame byte 0 (%MW26) e byte 1 (%MW27)
%MWr.m.c.24.6		Esplicito	BOOL	Cambia la modalità di gestione modem HALF/FULL DUPLEX. ● Se è impostato simultaneamente con RTS_ON (%MWr.m.c.24.10 funziona anche con RTS_OFF %MWr.m.c.24.11 e usa il DTR se è utilizzato 8 o .9) viene attivata la modalità modem half duplex. ● Se è impostato questo bit, ma nessuno dei segnali RTS/DTR (né %MWr.m.c.24.8, né %MWr.m.c.24.9, né %MWr.m.c.24.10, né %MWr.m.c.24.11) viene attivata la modalità full duplex. Il bit %MW26 è utilizzato per impostare l'inizio ritardo e %MW27 è utilizzato per impostare la fine ritardo. Pertanto, i bit %MW24.5, %MW24.1 e %MW24.2 non possono essere utilizzati simultaneamente. **NOTA**: è possibile che l'utente debba ripristinare lo stato corretto dei segnali RTS/DTR dopo l'accettazione del comando.
%MWr.m.c.24.7		Esplicito	BOOL	Salva l'indirizzo slave Modbus nella FLASH (%MW25).
%MWr.m.c.24.8	DTR_ON	Esplicito	BOOL	Imposta il segnale DTR (tensione positiva)
%MWr.m.c.24.9	DTR_OFF	Esplicito	BOOL	Reimposta il segnale DTR (tensione negativa)
%MWr.m.c.24.10		Esplicito	BOOL	Imposta il segnale RTS (tensione positiva)
%MWr.m.c.24.11		Esplicito	BOOL	Reimposta il segnale RTS (tensione negativa)

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.24.12	TO_MODBUS_ MASTER	Esplicito	BOOL	Passa alla modalità master
%MWr.m.c.24.13	TO_MODBUS_ SLAVE	Esplicito	BOOL	Passa alla modalità slave
%MWr.m.c.24.14	TO_CHAR_ MODE	Esplicito	BOOL	Passa alla modalità caratteri
%MWr.m.c.25		Esplicito	INT	Indirizzo dello slave Modbus da memorizzare nella FLASH
%MWr.m.c.26		Esplicito	INT	Nuovo EOF in modalità carattere (eq %KW6) se viene impostato %MW24.5: ● Bit 0: 1 byte 1 is set below, 0 no more byte 1 ● Bit 1: 1 aggiunge il byte 1, 0 non aggiunge il byte 1 ● Bit2..7 : deve essere nullo.HIGH BYTE: la fine frame byte 1 Inizio ritardo se %MW26.6 è impostato. Tempo di attesa dopo che il CTS è OK prima di iniziare ad inviare il frame. È utile per i modem che richiedono più tempo dopo il segnale CTS o che non gestiscono il segnale CTS (in questo caso il segnale RTS deve essere collegato al segnale CTS). Questo tempo è espresso in millisecondi, la precisione è di circa 3ms. Può essere eseguito solo in modalità RS232.
%MWr.m.c.27		Esplicito	INT	Nuovo EOF in modalità carattere (eq %KW7) se viene impostato %MW24.5: ● Bit 0: 1 byte 2 è impostato sotto, 0 più nessun byte 2 ● Bit 1: 1 aggiunge il byte 2, 0 non aggiunge il byte 2 ● Bit2..7 : deve essere nullo.HIGH BYTE: la fine frame byte 2 Inizio ritardo se %MW24.6 è impostato. Tempo di attesa dopo aver inviato un frame prima di rilasciare il segnale RTS, per lasciare al MODEM tempo sufficiente per completare l'invio del frame prima dell'interruzione del collegamento. Questo tempo è espresso in millisecondi, la precisione è di circa 3ms. Può essere eseguito solo in modalità RS232.
%MWr.m.c.28		Esplicito	INT	Riservato
%MWr.m.c.29		Esplicito	INT	Riservato

Indirizzo	Simbolo standard	Tipo di scambio	Tipo	Significato
%MWr.m.c.30		Esplicito	INT	silenziamento: ritardo tra i frame in millisecondi [0..10000]. Il valore utilizzato dipende dalla velocità. Se il valore è inferiore o superiore ai valori possibili, viene applicato il limite inferiore o il limite superiore e il comando viene accettato. Un valore 0 significa nessun silenziamento.
%MWr.m.c.31		Esplicito	INT	Riservato
%MWr.m.c.32		Esplicito	INT	Riservato

Esempio di codice

```
if HalfModemSlave then
 HalfModemMaster:=false;
  %MW0.9.0.24:=16#0440;(* passa alla modalità half duplex con RTS*)
  %MW0.9.0.26:=12;(* 12ms di attesa prima dell'invio quando CTS aumenta *)
 %MW0.9.0.27:=9; (* lascia RTS attivo 9ms dopo la fine dell'invio *)
  write_cmd(%ch0.9.0);(* invia il comando e i dati al canale NOM*)
end_if;
```

Sezione 8.4
Debug della comunicazione in modalità carattere

Schermata di debug della comunicazione in modalità carattere

Generale

La schermata di debug per la modalità caratteri è accessibile solo in modalità online.

Accesso alla schermata di debug

La tabella seguente descrive la procedura per accedere alla schermata di debug della comunicazione in modalità caratteri.

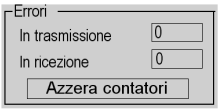
Passo	Azione
1	Accedere alla schermata di configurazione della comunicazione in modalità carattere. *(vedi pagina 172)*
2	Selezionare la scheda "Debug" sulla schermata visualizzata.

Descrizione della schermata Debug

La schermata di debug è composta dalla zona Errore e dalla zona Segnali (se RS232).

La zona Errore

La zona Errore è illustrata di seguito:



Questa zona indica il numero di interruzioni di comunicazione conteggiati dal modulo.

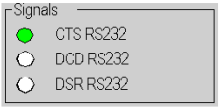
- **In trasmissione**: corrisponde al numero di interruzioni verificatisi durante la trasmissione (immagina della parola `%MW4`).
- **In ricezione**: corrisponde al numero di interruzioni verificatisi durante la ricezione (immagina della parola `%MW5`).

Il pulsante Azzera contatori azzera tutti i contatori.

La zona Segnali

Questa zona viene visualizzata solamente se RS232 viene selezionato nella schermata di configurazione. Se RS485 viene selezionato nella schermata di configurazione, questa finestra non viene visualizzata per nulla.

La zona Segnali è illustrata di seguito:



Questa zona indica l'attività dei segnali:

- **CTS RS232**: indica l'attività del segnale CTS.
- **DCD RS232**: indica l'attività del segnale DCD.
- **DSR RS232**: indica l'attività del segnale DSR.

Capitolo 9
Diagnostica Modulo BMX NOM 0200

Sezione 9.1
Diagnostica Modulo BMX NOM 0200

Oggetto della sezione

Questa sezione descrive l'aspetto diagnostico nell'implementazione di un modulo di comunicazione BMX NOM 0200.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Diagnostica di un modulo BMX NOM 0200	205
Diagnostica dettagliata del canale di comunicazione	207

Diagnostica di un modulo BMX NOM 0200

In breve

La funzione di diagnostica del modulo visualizza le anomalie quando si verificano, classificate in base alla loro categoria.

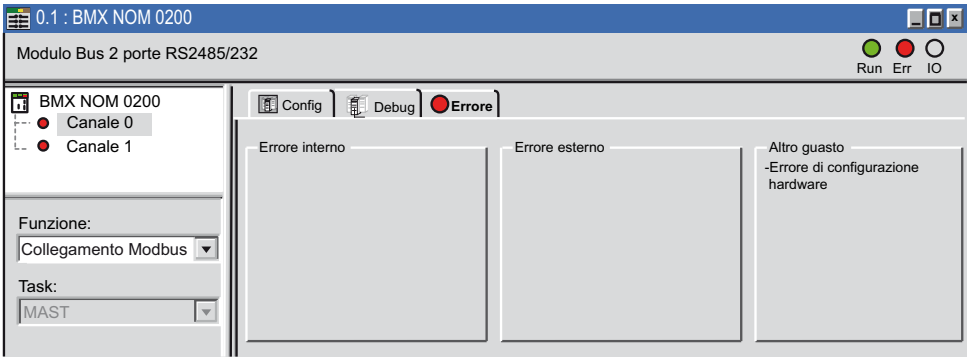
- **Errore rilevato internamente**:
  - evento del modulo
- **Evento esterno**:
  - controllo del cablaggio (filo interrotto, sovraccarico o cortocircuito)
- **Altre anomalie**:
  - canale non funzionante
  - anomalia nella configurazione
  - modulo assente o non alimentato

Un errore su un modulo è indicato da una serie di LED che diventano rossi, ad esempio:

- nell'editor di configurazione a livello del rack:
  - il LED del numero di rack
  - il LED del numero di slot del modulo nel rack
- nell'editor di configurazione a livello del modulo:
  - i LED **Err** e **I/O**, a seconda del tipo di errore rilevato
  - LED **Canale** nel campo **Canale**

Accesso alla schermata Diagnostica del modulo

La tabella seguente mostra la procedura per accedere alla schermata di diagnostica del modulo.

Passo	Azione
1	Aprire la schermata di debug del modulo.
2	Fare clic sul riferimento del modulo nell'area del canale e selezionare la scheda **Errore**. **Risultato**: viene visualizzato l'elenco degli errori rilevati del modulo.  **Note:** non è possibile accedere alla schermata di diagnostica del modulo se si verifica un errore di configurazione, un guasto grave o un errore rilevato per modulo mancante. Nella schermata viene visualizzato il seguente messaggio: "`Modulo assente o diverso da quello configurato per questa posizione`".

Elenco errori del modulo rilevati

Nella seguente tabella di riepilogo sono riportati i vari errori rilevati per un modulo di comunicazione:

Classificazione degli errori rilevati	Oggetti linguaggio
`Errore interno`: • `Errore rilevato del modulo`	• `%MWr.m.MOD.2.0`
`Errore esterno`: • `Morsettiera`	• `%MWr.m.MOD.2.2`
`Altro errore`: • `Canale(i) guasto(i)` • `Errore di configurazione hardware` • `Modulo assente o non alimentato`	• `%MWr.m.MOD.2.1` • `%MWr.m.MOD.2.5` • `%MWr.m.MOD.2.6`

Diagnostica dettagliata del canale di comunicazione

In breve

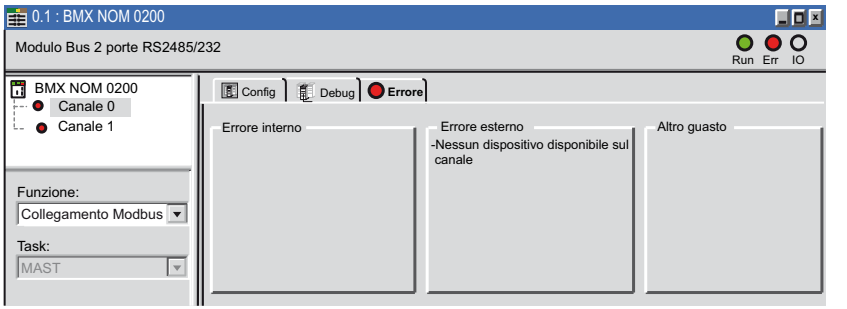
La funzione di diagnostica del canale visualizza gli errori rilevati nel momento in cui si verificano, classificati in base alla loro categoria:

- **errore rilevato internamente**
  - test automatici in corso
- **eventi esterni**
  - dispositivo assente
  - dispositivo non funzionante
  - timeout della comunicazione di collegamento seriale
- **Altri errori rilevati:**
  - errore dello strumento di linea
  - errore di configurazione
  - perdita della comunicazione
  - errore dell'applicazione

Il rilevamento di un canale in errore è indicato nella scheda **Debug** quando il LED, nella colonna **Errore** diventa rosso.

Accesso alla schermata Diagnostica del canale

La tabella seguente mostra la procedura per accedere alla schermata di diagnostica del canale.

Passo	Azione
1	Aprire la schermata di debug del modulo.
2	Per il canale non funzionante, fare clic sul pulsante situato nella colonna **Errore**. **Risultato**: viene visualizzato l'elenco degli errori rilevati del canale.  **Note:** le informazioni di diagnostica dei canali sono accessibili anche dal programma (istruzione READ_STS).

Elenco degli errori rilevati del canale

Nella seguente tabella di riepilogo sono riportati i vari errori rilevati per un collegamento configurato in seriale:

Classificazione degli errori rilevati	Oggetti linguaggio
Errore interno: ● Test automatici in corso	● %MWr.m.c.2.4
Errore esterno: ● Nessun dispositivo disponibile sul canale ● Errore del dispositivo ● Errore di timeout (CTS)	● %MWr.m.c.2.0 ● %MWr.m.c.2.1 ● %MWr.m.c.2.3
Altro errore: ● Errore linea utensile ● Errore di configurazione hardware ● Problema di comunicazione con il PLC ● Errore applicazione	● %MWr.m.c.2.2 ● %MWr.m.c.2.5 ● %MWr.m.c.2.6 ● %MWr.m.c.2.7

Capitolo 10
Oggetti linguaggio di comunicazioni Modbus e modalità caratteri

Oggetto di questo capitolo

Questo capitolo descrive gli oggetti linguaggio associati alle comunicazioni Modbus e alla modalità caratteri e le diverse modalità di utilizzo.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
10.1	Oggetti linguaggio e IODDT delle comunicazioni Modbus e in modalità carattere	210
10.2	Oggetti linguaggio e IODDT generici per i protocolli di comunicazione	218
10.3	Oggetti linguaggio e IODDT associati alla comunicazione Modbus	222
10.4	Oggetti linguaggio e IODDT associati alla comunicazione in modalità caratteri	230
10.5	IODDT tipo T_GEN_MOD applicabili a tutti i moduli	238
10.6	Oggetti linguaggio e DDT dei dispositivi associati alla comunicazione Modbus	240

Sezione 10.1
Oggetti linguaggio e IODDT delle comunicazioni Modbus e in modalità carattere

Argomento della sezione

Questa sezione fornisce una panoramica dei punti generali relativi agli IODDT e agli oggetti linguaggio per le comunicazioni Modbus e in modalità carattere.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Introduzione agli oggetti linguaggio delle comunicazioni Modbus e in modalità carattere	211
Oggetti linguaggio a scambio implicito associati alla funzione specifica dell'applicazione	212
Oggetti di linguaggio a scambio esplicito associati alla funzione specifica all'applicazione	213
Gestione degli scambi e dei rapporti con oggetti espliciti	215

Introduzione agli oggetti linguaggio delle comunicazioni Modbus e in modalità carattere

Generale

Gli IODDT sono predefiniti dal produttore. Essi contengono oggetti linguaggio di tipo ingresso/uscita relativi al canale di un modulo specifico dell'applicazione.

Le comunicazioni Modbus e in modalità carattere presentano tre IODDT associati:

- `T_COM_STS_GEN`, che si applica ai protocolli di comunicazione, ad eccezione di Fipio e Ethernet.
- `T_COM_MB_BMX`, che è specifico della comunicazione Modbus;
- `T_COM_CHAR_BMX`, che è specifico della comunicazione in modalità carattere.

NOTA: è possibile creare variabili IODDT in due modi diversi:

- utilizzando la scheda Oggetti I/O *(vedi Unity Pro, Modalità operative)*
- tramite l'Editor dati.

Tipi di oggetti linguaggio

Ciascun IODDT contiene un gruppo di oggetti linguaggio utilizzato per controllarne il funzionamento.

Esistono due tipi di oggetti di linguaggio:

- Oggetti di scambio implicito: Tali oggetti vengono scambiati automaticamente ad ogni ciclo del task associato al processore.
- Oggetti di scambio esplicito: tali oggetti vengono scambiati su richiesta dell'applicazione, tramite istruzioni di scambio esplicite.

Gli scambi impliciti riguardano lo stato dei processori, i segnali di comunicazione, gli slave, ecc.

Gli scambi espliciti vengono utilizzati per definire le impostazioni del processore ed eseguire operazioni di diagnostica.

Oggetti linguaggio a scambio implicito associati alla funzione specifica dell'applicazione

In breve

L'utilizzo di un'interfaccia integrata, specifica dell'applicazione, o l'aggiunta di un modulo migliora automaticamente l'applicazione di oggetti linguaggio usata per programmare questa interfaccia o il modulo.

Questi oggetti corrispondono alle immagini di I/O e alle informazioni software del modulo o dell'interfaccia specifica integrata.

Promemoria

Gli ingressi del modulo (`%I` e `%IW`) vengono aggiornati nella memoria del PLC all'inizio del task, o quando il PLC è in modalità RUN o STOP.

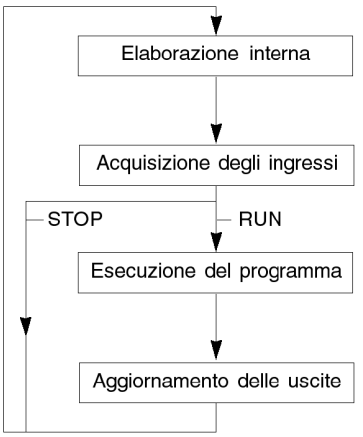
Le uscite (`%Q` e `%QW`) vengono aggiornate alla fine del task, solo quando il PLC è in modalità RUN.

NOTA: quando il task è in modalità STOP, a seconda della configurazione selezionata sono possibili una delle due eventualità:

- le uscite vengono messe in posizione di sicurezza (modalità posizione di sicurezza).
- le uscite mantengono l'ultimo valore (modalità mantieni).

Illustrazione

Il grafico riportato di seguito illustra il ciclo di funzionamento relativo a un task di PLC (esecuzione ciclica):



Oggetti di linguaggio a scambio esplicito associati alla funzione specifica all'applicazione

In breve

Gli scambi espliciti sono scambi effettuati su richiesta del programma utente che utilizzano le seguenti istruzioni:

- READ_STS *(vedi Unity Pro, Gestione I/O, Libreria dei blocchi funzione)*: lettura parole di stato
- WRITE_CMD *(vedi Unity Pro, Gestione I/O, Libreria dei blocchi funzione)*: scrittura parola di comando

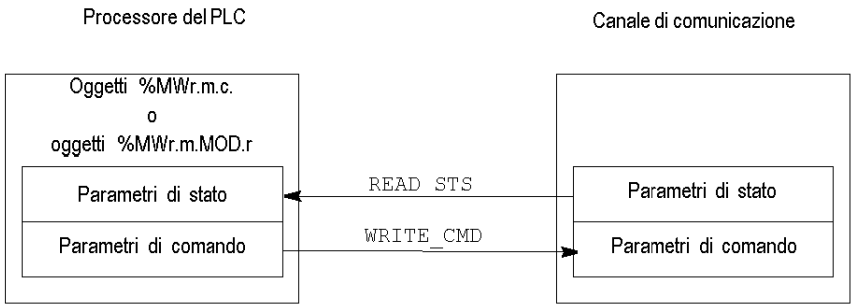
Tali scambi si applicano a una serie di oggetti `%MW` dello stesso tipo (stati, comandi o parametri) appartenenti a un canale.

NOTA: Gli oggetti forniscono informazioni riguardo al processore o al modulo, possono essere utilizzati per comandarli (es.: comando switch) e per definirne le modalità si funzionamento (salvataggio e ripristino dei parametri di regolazione nel processo dell'applicazione).

NOTA: Le istruzioni `READ_STS` e `WRITE_CMD` vengono eseguite contemporaneamente al task che le chiama e sempre senza errore. Il risultato di tali istruzioni è immediatamente disponibile dopo la loro esecuzione.

Principio generale per l'uso delle istruzioni esplicite

Lo schema seguente illustra i diversi tipi di scambi espliciti effettuabili tra il processore e il canale di comunicazione.



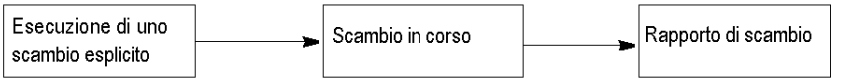
Gestione degli scambi

Durante uno scambio esplicito, occorre controllarne lo svolgimento per prendere in considerazione i dati solo quando lo scambio è stato effettuato correttamente.

A tale scopo, sono disponibili due tipi di informazioni:

- informazioni relative allo scambio in corso; *(vedi Unity Pro, Gestione I/O, Libreria dei blocchi funzione)*
- rapporto relativo allo scambio. *(vedi Unity Pro, Gestione I/O, Libreria dei blocchi funzione)*

Lo schema seguente descrive il principio di gestione di uno scambio:



NOTA: al fine di evitare più scambi espliciti in uno stesso momento per lo stesso canale, è necessario effettuare il test del valore della parola EXCH_STS (`%MWr.m.c.0`) dell'IODDT associato al canale prima di richiamare qualsiasi EF con questo canale.

Gestione degli scambi e dei rapporti con oggetti espliciti

In breve

Quando i dati vengono scambiati tra la memoria del PLC e il modulo, è possibile che quest'ultimo richieda diversi cicli di task per riconoscere tali informazioni.

Per gestire gli scambi, tutti gli IODDT dispongono di due parole:

- `EXCH_STS` (`%MWr.m.c.0`): scambio in corso
- `EXCH_RPT` (`%MWr.m.c.1`): rapporto.

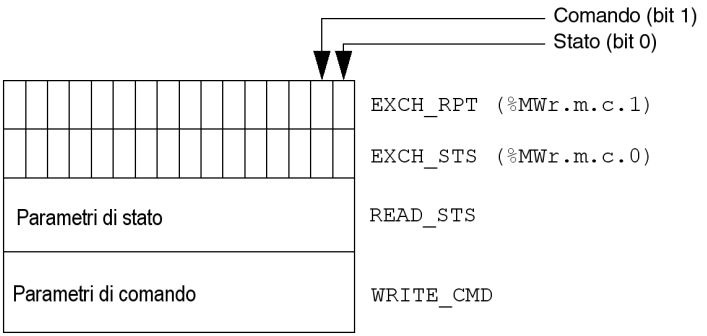
NOTA:

A seconda della posizione del modulo, la gestione degli scambi espliciti (ad esempio, `%MW0.0.MOD.0.0`) non verrà rilevata dall'applicazione:

- per i moduli all'interno del rack, gli scambi espliciti vengono eseguiti immediatamente sul bus PLC locale e terminati prima della fine del task di esecuzione. Ad esempio,`READ_STS` è sempre terminato quando il bit `%MW0.0.mod.0.0` viene controllato dall'applicazione.
- per il bus remoto (Fipio ad esempio), gli scambi espliciti non sono sincroni con il task di esecuzione, in modo da rendere possibile il rilevamento da parte dell'applicazione.

Illustrazione

Nella figura seguente sono mostrati i vari bit significativi per la gestione degli scambi:



Descrizione dei bit significativi

Ciascun bit delle parole EXCH_STS (%MWr.m.c.0) e EXCH_RPT (%MWr.m.c.1) è associato ad un tipo di parametro:

- I bit di rango 0 sono associati ai parametri di stato:
  - Il bit STS_IN_PROGR (%MWr.m.c.0.0) indica se è in corso una richiesta di lettura delle parole di stato.
  - Il bit STS_ERR (%MWr.m.c.1.0) specifica se una richiesta di lettura delle parole di stato è accettata dal canale del modulo.
- I bit di rango 1 sono associati ai parametri di comando:
  - Il bit CMD_IN_PROGR (%MWr.m.c.0.1) indica se è in corso l'invio dei parametri di comando al canale del modulo.
  - Il bit CMD_ERR (%MWr.m.c.1.1) indica se i parametri di comando sono accettati o meno dal canale del modulo.

NOTA: r rappresenta il numero di rack e m la posizione del modulo nel rack, mentre crappresenta il numero di canale nel modulo.

NOTA: Le parole di scambio e rapporto esistono anche a livello del modulo EXCH_STS (%MWr.m.MOD.0) e EXCH_RPT (%MWr.m.MOD.1) come per gli IODDT di tipo T_GEN_MOD .

Flag di esecuzione di uno scambio esplicito: EXCH_STS

La seguente tabella mostra i bit di controllo degli scambi espliciti della parola EXCH_STS (%MWr.m.c.0):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_IN_PROGR	BOOL	R	Lettura delle parole di stato del canale in corso	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Scambio di parametri di comando in corso	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Scambio dei parametri di regolazione in corso	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Riconfigurazione del modulo in corso	%MWr.m.c.0.15

NOTA: Se il modulo non è presente o è scollegato, gli oggetti di scambio esplicito (READ_STS, ad esempio) non vengono inviati al processore (STS_IN_PROG (%MWr.m.c.0.0) = 0), ma le parole vengono aggiornate.

Rapporto di scambio esplicito: EXCH_RPT

La tabella seguente mostra i bit di rapporto della parola `EXCH_RPT` (`%MWr.m.c.1`):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_ERR	BOOL	R	Errore rilevato durante la lettura delle parole di stato del canale (1 = guasto rilevato)	`%MWr.m.c.1.0`
CMD_ERR	BOOL	R	Errore rilevato durante uno scambio del parametro di comando (1 = guasto rilevato)	`%MWr.m.c.1.1`
ADJ_ERR	BOOL	R	Interruzioni durante lo scambio dei parametri di regolazione (1 = guasto rilevato)	`%MWr.m.c.1.2`
RECONF_ERR	BOOL	R	Interruzioni durante la riconfigurazione del canale (1 = guasto rilevato)	`%MWr.m.c.1.15`

Sezione 10.2
Oggetti linguaggio e IODDT generici per i protocolli di comunicazione

Argomento della sezione

Questa sezione presenta gli oggetti linguaggio e gli IODDT generici applicabili a tutti i protocolli di comunicazione, ad eccezione di Fipio e Ethernet.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Dettagli degli oggetti di scambi impliciti IODDT di tipo T_COM_STS_GEN	219
Dettagli degli oggetti di scambi espliciti IODDT di tipo T_COM_STS_GEN	220

Dettagli degli oggetti di scambi impliciti IODDT di tipo T_COM_STS_GEN

In breve

La tabella seguente presenta gli oggetti di scambio implicito IODDT di tipo `T_COM_STS_GEN` applicabili a tutti i protocolli di comunicazione ad eccezione di Fipio.

Bit di errore

La tabella seguente mostra il significato del bit di errore `CH_ERROR` (%Ir.m.c.ERR):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
CH_ERROR	EBOOL	R	Bit di errore canale di comunicazione.	%Ir.m.c.ERR

Dettagli degli oggetti di scambi espliciti IODDT di tipo T_COM_STS_GEN

In breve

In questa sezione sono descritti gli oggetti di scambio esplicito IODDT di tipo `T_COM_STS_GEN` applicabili a tutti i protocolli di comunicazione ad eccezione di Fipio ed Ethernet. Include gli oggetti di tipo parola formati da bit che hanno un significato specifico. Questi oggetti sono illustrati in modo dettagliato di seguito.

In questa parte, la variabile `IODDT_VAR1` è di tipo `T_COM_STS_GEN`.

Osservazioni

In generale, il significato dei bit viene fornito per lo stato 1 del bit. In casi specifici, si fornisce una spiegazione per ciascuno stato del bit.

Non tutti i bit vengono utilizzati.

Flag di esecuzione di uno scambio esplicito: EXCH_STS

Nella seguente tabella è riportato il significato dei bit di controllo di scambio del canale `EXCH_STS` (`%MWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
STS_IN_PROGR	BOOL	R	Lettura parole di stato del canale in corso.	`%MWr.m.c.0.0`
CMD_IN_PROGR	BOOL	R	Scambio del parametro di comando in corso.	`%MWr.m.c.0.1`

Rapporto di scambio esplicito: EXCH_RPT

Nella seguente tabella è riportato il significato dei bit del rapporto di scambio `EXCH_RPT` (`%MWr.m.c.1`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
STS_ERR	BOOL	R	Errore rilevato di lettura delle parole di stato del canale.	`%MWr.m.c.1.0`
CMD_ERR	BOOL	R	Errore rilevato durante lo scambio del parametro di comando.	`%MWr.m.c.1.1`

Errori canale standard: CH_FLT

La tabella seguente mostra il significato dei bit della parola di stato `CH_FLT` (`%MWr.m.c.2`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
NO_DEVICE	BOOL	R	Nessun dispositivo operativo sul canale.	`%MWr.m.c.2.0`
ONE_DEVICE_FLT	BOOL	R	Un dispositivo del canale è inoperativo.	`%MWr.m.c.2.1`
BLK	BOOL	R	Morsettiera non collegata.	`%MWr.m.c.2.2`
TO_ERR	BOOL	R	Temporizzazione superata (analisi necessaria).	`%MWr.m.c.2.3`
INTERNAL_FLT	BOOL	R	Errore interno rilevato o test automatico del canale.	`%MWr.m.c.2.4`
CONF_FLT	BOOL	R	Configurazioni hardware e software diverse.	`%MWr.m.c.2.5`
COM_FLT	BOOL	R	Analisi di comunicazione necessaria con il canale.	`%MWr.m.c.2.6`
APPLI_FLT	BOOL	R	Errore dell'applicazione rilevato (regolazione o configurazione).	`%MWr.m.c.2.7`

La lettura viene eseguita tramite l'istruzione `READ_STS` (`IODDT_VAR1`).

Sezione 10.3
Oggetti linguaggio e IODDT associati alla comunicazione Modbus

Argomento della sezione

In questa sezione vengono descritti gli oggetti linguaggio e gli IODDT associati alla comunicazione Modbus.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Dettagli concernenti oggetti di linguaggio a scambi espliciti per una funzione Modbus	223
Dettagli sugli oggetti di scambio implicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT	224
Dettagli sugli oggetti di scambio esplicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT	225
Dettagli del linguaggio oggetti associato alla configurazione in modalità Modbus	228

Dettagli concernenti oggetti di linguaggio a scambi espliciti per una funzione Modbus

In breve

La tabella seguente mostra gli oggetti linguaggio per comunicazioni Modbus in modalità master o slave. Tali oggetti non sono integrati negli IODDT.

Elenco di oggetti di scambio esplicito in modalità master o slave

La tabella seguente mostra gli oggetti di scambio esplicito:

Indirizzo	Tipo	Accesso	Significato
%MWr.m.c.4	INT	R	Numero di risposte ricevute correttamente.
%MWr.m.c.5	INT	R	Numero delle risposte ricevute con errore CRC.
%MWr.m.c.6	INT	R	Numero delle risposte ricevute con codice di eccezione in modalità slave.
%MWr.m.c.7	INT	R	Numero di messaggi inviati in modalità slave.
%MWr.m.c.8	INT	R	Numero di messaggi inviati senza risposta in modalità slave.
%MWr.m.c.9	INT	R	Numero delle risposte ricevute con riconoscimento negativo.
%MWr.m.c.10	INT	R	Numero di messaggi ripetuti in modalità slave.
%MWr.m.c.11	INT	R	Numero di errori carattere rilevati.
%MWr.m.c.24.0	BOOL	RW	Azzeramento dei contatori d’errore rilevati.

Dettagli sugli oggetti di scambio implicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT

In breve

Nella seguente tabella sono riportati gli oggetti di scambio implicito degli IODDT di tipo `T_COM_MB_BMX` e `T_COM_MB_BMX_CONF_EXT` utilizzabili per la comunicazione seriale Modbus. Questi oggetti differiscono in termini di **disponibilità di configurazione degli oggetti** *(vedi pagina 227)*.

Bit CH_ERROR

La tabella seguente illustra il significato del bit di errore `CH_ERROR` (`%Ir.m.c.ERR`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
CH_ERROR	EBOOL	R	Bit di errore del canale di comunicazione	`%Ir.m.c.ERR`

Oggetti parola in modalità Modbus master

La tabella seguente mostra il significato del bit della parola `INPUT_SIGNALS` (`%IWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
DCD	BOOL	R	Il vettore dati rileva un segnale RS232 (applicabile solo al modulo BMX NOM 0200)	`%IWr.m.c.0.0`
CTS	BOOL	R	Segnale RS232 Clear to send	`%IWr.m.c.0.2`
DSR	BOOL	R	Segnale RS232 Data set ready (applicabile solo al modulo BMX NOM 0200)	`%IWr.m.c.0.3`

NOTA: Quando CTS è verde in Punit, significa che `%IWr.m.c.0.0` è su 1 e che la tensione su questo segnale è positiva. Applicabile anche a DCD e DSR.

Oggetto parola in modalità Modbus slave

Gli oggetti linguaggio sono uguali a quelli della funzione Modbus master. Solo gli oggetti nella tabella seguente sono differenti.

La tabella seguente mostra il significato del bit della parola `INPUT_SIGNALS` (`%IWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
LISTEN_ONLY	BOOL	R	Modalità di solo ascolto:	`%IWr.m.c.0.8`

Dettagli sugli oggetti di scambio esplicito IODDT di tipo T_COM_MB_BMX e T_COM_MB_BMX_CONF_EXT

In breve

Questa sezione presenta gli oggetti di scambio esplicito dello IODDT di tipo `T_COM_MB_BMX` e `T_COM_MB_BMX_CONF_EXT` relativi alla comunicazione Modbus seriale e che differiscono in termini di **disponibilità di configurazione degli oggetti** *(vedi pagina 227)*. Include gli oggetti di tipo parola formati da bit che hanno un significato specifico. Questi oggetti sono illustrati in modo dettagliato di seguito.

In questa parte, la variabile `IODDT_VAR1` è di tipo `T_COM_STS_GEN`.

Osservazioni

In generale, il significato dei bit viene fornito per lo stato 1 del bit. In casi specifici, si fornisce una spiegazione per ciascuno stato del bit.

Non tutti i bit sono utilizzati.

Flag di esecuzione di uno scambio esplicito: EXCH_STS

La tabella seguente illustra i significati dei bit di controllo di scambio del canale `EXCH_STS` (`%MWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_IN_PROGR	BOOL	R	Lettura parole di stato del canale in corso.	`%MWr.m.c.0.0`
CMD_IN_PROGR	BOOL	R	Scambio del parametro di comando in corso.	`%MWr.m.c.0.1`
ADJ_IN_PROGR	BOOL	R	Scambio parametri di regolazione in corso (non applicabile al modulo BMX NOM 0200).	`%MWr.m.c.0.2`

Rapporto di scambio esplicito: EXCH_RPT

Nella seguente tabella sono riportati i vari significati dei bit del rapporto di scambio `EXCH_RPT` (`%MWr.m.c.1`):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_ERR	BOOL	R	Errore rilevato di lettura delle parole di stato del canale.	`%MWr.m.c.1.0`
CMD_ERR	BOOL	R	Anomalia durante lo scambio del parametro di comando.	`%MWr.m.c.1.1`
ADJ_ERR	BOOL	R	Anomalia durante lo scambio dei parametri di regolazione (non applicabile al modulo BMX NOM 0200).	`%MWr.m.c.1.2`

Errori rilevati del canale standard: CH_FLT

La tabella seguente illustra i vari significati dei bit della parola di stato CH_FLT (%MWr.m.c.2):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
NO_DEVICE	BOOL	R	Nessun dispositivo operativo sul canale.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Un dispositivo del canale è inoperativo.	%MWr.m.c.2.1
BLK	BOOL	R	Morsettiera non collegata.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Temporizzazione superata (analisi necessaria).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Errore interno rilevato o test automatico del canale.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configurazioni hardware e software diverse.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Analisi di comunicazione necessaria con il canale.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Errore dell'applicazione rilevato (errore di regolazione o configurazione).	%MWr.m.c.2.7

La lettura viene eseguita tramite l'istruzione READ_STS (IODDT_VAR1).

Stato canale specifico: %MWr.m.c.3

La tabella seguente illustra i vari significati dei bit della parola di stato del canale PROTOCOL (%MWr.m.c.3):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
PROTOCOL	INT	R	Byte 0 = 16#06 per funzione Modbus master.	%MWr.m.c.3
PROTOCOL	INT	R	Byte 0 = 16#07 per funzione Modbus slave.	%MWr.m.c.3

La lettura viene eseguita tramite l'istruzione READ_STS (IODDT_VAR1).

Comando canale: %MWr.m.c.24

La tabella seguente illustra i vari significati dei bit della parola CONTROL (%MWr.m.c.24):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
DTR_ON	BOOL	R/W	Impostare il segnale Terminale dati pronto.	%MWr.m.c.24.8
DTR_OFF	BOOL	R/W	Reimpostare il segnale Terminale dati pronto.	%MWr.m.c.24.9
TO_MODBUS_MASTER	BOOL	R/W	Passaggio dalla modalità carattere o Modbus slave a Modbus master.	%MWr.m.c.24.12
TO_MODBUS_SLAVE	BOOL	R/W	Passaggio dalla modalità carattere o Modbus master a Modbus slave.	%MWr.m.c.24.13
TO_CHAR_MODE	BOOL	R/W	Passaggio da Modbus a modalità carattere.	%MWr.m.c.24.14

Il comando viene eseguito con l'istruzione WRITE_CMD (IODDT_VAR1).

Per ulteriori informazioni sulla modalità di cambiamento dei protocolli, fare riferimento a **cambi di protocollo** *(vedi pagina 246)*.

Oggetti configurazione esterna di tipo T_COM_MB_BMX_CONF_EXT: %MWr.m.c.24.7 e %MWr.m.c.25

La tabella di seguito illustra il significato del bit CONTROL (%MWr.m.c.24.7) e della parola CONTROL_DATA (%MWr.m.c.25) specificatamente intese per la programmazione del modulo BMX NOM 0200:

Simbolo standard	Tipo	Accesso	Significato	Riferimento
SAVE_CTRL_DATA	BOOL	R/W	Salvare i dati di controllo nella memoria FLASH	%MWr.m.c.24.7
CONTROL_DATA	BOOL	R/W	Indirizzo slave Modbus da archiviare nella memoria FLASH.	%MWr.m.c.25

Dettagli del linguaggio oggetti associato alla configurazione in modalità Modbus

In breve

Le tabelle seguenti mostrano tutti gli oggetti linguaggio di configurazione per comunicazione in modalità Modbus. Questi oggetti non sono integrati nei moduli IODDT e possono essere visualizzati dal programma applicativo.

Elenco di oggetti di scambio esplicito per la modalità Master

Nella seguente tabella sono indicati gli oggetti di scambio esplicito.

Indirizzo	Tipo	Accesso	Significato
%KWr.m.c.0	INT	R	Il byte 0 di questa parola corrisponde al tipo: ● il valore 6 corrisponde al master ● il valore 7 corrisponde allo slave
%KWr.m.c.1	INT	R	Il byte 0 di questa parola corrisponde alla velocità di trasmissione. Questo byte può assumere più valori: ● il valore -2 (0xFE) corrisponde a 300 bit/s ● il valore -1 (0xFF) corrisponde a 600 bit/s ● il valore 0 (0x00) corrisponde a 1200 bit/s ● il valore 1 (0x01) corrisponde a 2400 bit/s ● il valore 2 (0x02) corrisponde a 4800 bit/s ● il valore 3 (0x03) corrisponde a 9600 bit/s ● il valore 4 (0x04) corrisponde a 19200 bit/s (valore predefinito) ● il valore 5 (0x05) corrisponde a 38400 bit/s ● il valore 6 (0x06) corrisponde a 57600 bit/s (applicabile solamente al modulo BMX NOM 0200) ● il valore 7 (0x07) corrisponde a 115200 bit/s (applicabile solamente al modulo BMX NOM 0200) Il byte 1 di questa parola corrisponde al formato: ● bit 8: numero di bit (1 = 8 bit (RTU), 0 = 7 bit (ASCII)) ● bit 9 = 1: gestione della parità (1 = con, 0 = senza) ● bit 10: tipo di parità (1 = dispari, 0 = pari) ● bit 11: numero di bit di stop (1 = 1 bit, 0 = 2 bit) ● bit 13: linea fisica (1 = RS232, 0 = RS485) ● bit 14: Segnali del modulo DTR/DSR/DCD (applicabili al modulo BMX NOM 0200 e solo per la linea fisica RS232). Se il bit è impostato su 1, vengono gestiti i segnali del modem. ● bit 15: Segnali di gestione del flusso hardware RTS/CTS. Se viene selezionato RS232 questo bit può assumere 2 valori diversi: 0 per RX/TX e 1 per RX/TX + RTS/CTS. Se viene selezionato RS485 il valore predefinito è 0 e corrisponde a RX/TX.
%KWr.m.c.2	INT	R	Ritardo tra frame (solo in modalità RTU): valori in ms tra 2 e 10000 ms (dipendono dalla velocità di trasmissione e dal formato selezionati). Il valore predefinito è 2 ms se è selezionata la casella di controllo Predefinito. 10 s corrisponde a un'attesa infinita.

Indirizzo	Tipo	Accesso	Significato
%KWr.m.c.3	INT	R	Nella modalità Modbus Master questo oggetto corrisponde al ritardo di risposta, in ms, da 10 ms a 1000 ms. 100 ms è il valore predefinito. 10 s corrisponde a un'attesa infinita.
%KWr.m.c.4	INT	R	Disponibile solo in modalità Modbus Master. Il byte 0 di questa parola è il numero di tentativi da 0 a 15. Il valore predefinito è 3.
%KWr.m.c.5	INT	R	Se è selezionato RS232, questa parola corrisponde al tempo di ritardo RTS/CTS in centesimi di ms da 0 a 100. Se è selezionato RS485, il valore predefinito è 0.

Elenco di oggetti di scambio esplicito per la modalità Slave

Gli oggetti linguaggio per la funzione slave di Modbus sono identici a quelli della funzione master di Modbus. L'unica differenza riguarda gli oggetti seguenti:

Indirizzo	Tipo	Accesso	Significato
%KWr.m.c.3	INT	R	Nella modalità Modbus Slave il byte 0 di questo oggetto corrisponde al numero slave [0/1, 247]. Per il modulo BMX NOM 0200, il valore 0 significa che il numero dello slave è codificato nella memoria FLASH
%KWr.m.c.4	INT	R	Utilizzato solo in modalità Modbus Master.

Sezione 10.4
Oggetti linguaggio e IODDT associati alla comunicazione in modalità caratteri

Argomento della sezione

In questa sezione vengono descritti gli oggetti linguaggio e gli IODDT associati alla comunicazione in modalità carattere.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Dettagli concernenti oggetti di linguaggio a scambi espliciti per comunicazione in modalità caratteri	231
Dettagli relativi agli oggetti di scambio implicito dello IODDT di tipo T_COM_CHAR_BMX	232
Dettagli degli oggetti di scambio esplicito IODDT di Tipo T_COM_CHAR_BMX	233
Dettagli del linguaggio oggetti associato alla configurazione in modalità carattere	236

Dettagli concernenti oggetti di linguaggio a scambi espliciti per comunicazione in modalità caratteri

In breve

La tabella seguente mostra tutti gli oggetti linguaggio di configurazione per comunicazione in modalità carattere. Tali oggetti non sono integrati negli IODDT.

Elenco di oggetti di scambio esplicito

La tabella seguente mostra gli oggetti di scambio esplicito:

Indirizzo	Tipo	Accesso	Significato
%MWr.m.c.4	INT	R	Anomalia nei caratteri trasmessi.
%MWr.m.c.5	INT	R	Anomalia nei caratteri ricevuti.
%MWr.m.c.24.0	BOOL	RW	Azzera contatori d'errore quando il bit è impostato a 1.
%QWr.m.c.0 = 16#DEAD	INT	RW	Riavviare BMX NOM 0200.

Dettagli relativi agli oggetti di scambio implicito dello IODDT di tipo T_COM_CHAR_BMX

In breve

Nella seguente tabella vengono riportati gli oggetti di scambio implicito dello IODDT di tipo `T_COM_CHAR_BMX` relativo alla comunicazione in modalità carattere.

Bit di errore

La tabella seguente illustra il significato del bit di errore `CH_ERROR` (`%Ir.m.c.ERR`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
CH_ERROR	EBOOL	R	Bit di errore canale di comunicazione.	`%Ir.m.c.ERR`

Oggetto segnale all'ingresso

La tabella seguente mostra il significato del bit della parola `INPUT_SIGNALS` (`%IWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
DCD	BOOL	R	Portante dati ha rilevato il segnale RS232 (applicabile solo al modulo BMX NOM 0200).	`%IWr.m.c.0.0`
CTS	BOOL	R	Pronto a trasmettere il segnale RS232.	`%IWr.m.c.0.2`
DSR	BOOL	R	Data set ready ha rilevato il segnale RS232 (applicabile solo al modulo BMX NOM 0200).	`%IWr.m.c.0.3`

NOTA: Quando CTS è verde in Punit, significa che `%IWr.m.c.0.0` è su 1 e che la tensione su questo segnale è positiva. È anche applicabile a DCD e DSR.

Dettagli degli oggetti di scambio esplicito IODDT di Tipo T_COM_CHAR_BMX

In breve

Nella seguente tabella vengono riportati gli oggetti di scambio esplicito dello IODDT di tipo `T_COM_CHAR_BMX` relativo alla comunicazione in modalità carattere. Include gli oggetti di tipo parola formati da bit che hanno un significato specifico. Questi oggetti sono illustrati in modo dettagliato di seguito.

In questa parte, la variabile `IODDT_VAR1` è di tipo `T_COM_STS_GEN`.

Osservazioni

In generale, il significato dei bit viene fornito per lo stato 1 del bit. In casi specifici, si fornisce una spiegazione per ciascuno stato del bit.

Non tutti i bit sono utilizzati.

Flag di esecuzione di uno scambio esplicito: EXCH_STS

La tabella seguente illustra i significati dei bit di controllo di scambio del canale `EXCH_STS` (`%MWr.m.c.0`):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_IN_PROGR	BOOL	R	Lettura delle parole di stato del canale in corso.	`%MWr.m.c.0.0`
CMD_IN_PROGR	BOOL	R	Scambio del parametro di comando in corso.	`%MWr.m.c.0.1`
ADJ_IN_PROGR	BOOL	R	Scambio parametri di regolazione in corso (non applicabile al modulo BMX NOM 0200).	`%MWr.m.c.0.2`

Rapporto di scambio esplicito: EXCH_RPT

Nella seguente tabella è riportato il significato dei bit del rapporto di scambio `EXCH_RPT` (`%MWr.m.c.1`):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STS_ERR	BOOL	R	Errore rilevato di lettura delle parole di stato del canale.	`%MWr.m.c.1.0`
CMD_ERR	BOOL	R	Anomalia durante lo scambio del parametro di comando.	`%MWr.m.c.1.1`
ADJ_ERR	BOOL	R	Anomalia durante lo scambio dei parametri di regolazione (non applicabile al modulo BMX NOM 0200).	`%MWr.m.c.1.2`

Errori standard rilevati nel canale, CH_FLT

La tabella seguente illustra i vari significati dei bit della parola di stato CH_FLT (%MWr.m.c.2):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
NO_DEVICE	BOOL	R	Nessun dispositivo operativo sul canale.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Un dispositivo del canale è inoperativo.	%MWr.m.c.2.1
BLK	BOOL	R	Morsettiera non collegata.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Temporizzazione superata (analisi necessaria).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Errore interno rilevato o test automatico del canale.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configurazioni hardware e software diverse.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Analisi di comunicazione necessaria con il PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Errore dell'applicazione rilevato (errore di regolazione o configurazione).	%MWr.m.c.2.7

La lettura viene eseguita tramite l'istruzione READ_STS (IODDT_VAR1).

Stato del canale specifico, %MWr.m.c.3

La tabella seguente illustra i vari significati dei bit della parola di stato del canale PROTOCOL (%MWr.m.c.3):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
PROTOCOL	INT	R	Byte 0 = 16#03 per funzione modalità carattere.	%MWr.m.c.3

La lettura viene eseguita tramite l'istruzione READ_STS (IODDT_VAR1).

Comando del canale %MWr.m.c.24

La tabella seguente illustra i vari significati dei bit della parola CONTROL (%MWr.m.c.24):

Simbolo standard	Tipo	Accesso	Significato	Riferimento
DTR_ON	BOOL	R/W	Impostare il segnale Terminale dati pronto.	%MWr.m.c.24.8
DTR_OFF	BOOL	R/W	Reimpostare il segnale Terminale dati pronto.	%MWr.m.c.24.9

Il comando viene eseguito con l'istruzione WRITE_CMD (IODDT_VAR1).

Per ulteriori informazioni sulla modalità di cambiamento dei protocolli, fare riferimento a cambi di protocollo *(vedi pagina 246)*.

Oggetto della parola %QWr.m.c.0

La tabella seguente presenta il significato del bit 0 della parola %QWr.m.c.0:

Simbolo standard	Tipo	Accesso	Significato	Riferimento
STOP_EXCH	BOOL	R/W	Interrompe tutte le modifiche sul fronte di salita (disponibile solo sul modulo BMX NOM 0200).	%QWr.m.c.0.0

Dettagli del linguaggio oggetti associato alla configurazione in modalità carattere

In breve

La tabella seguente presenta tutti gli oggetti linguaggio di configurazione per comunicazione in modalità carattere. Questi oggetti non sono integrati nei moduli IODDT e possono essere visualizzati dal programma applicativo.

Elenco di oggetti di scambio esplicito per la modalità carattere

Nella seguente tabella sono indicati gli oggetti di scambio esplicito.

Indirizzo	Tipo	Accesso	Significato
%KWr.m.c.0	INT	R	Il byte 0 di questa parola corrisponde al tipo. Il valore 3 corrisponde alla modalità carattere.
%KWr.m.c.1	INT	R	Il byte 0 di questa parola corrisponde alla velocità di trasmissione. Questo byte può assumere più valori: ● il valore -2 (0xFE) corrisponde a 300 bit/s ● il valore -1 (0xFF) corrisponde a 600 bit/s ● il valore 0 (0x00) corrisponde a 1200 bit/s ● il valore 1 (0x01) corrisponde a 2400 bit/s ● il valore 2 (0x02) corrisponde a 4800 bit/s ● il valore 3 (0x03) corrisponde a 9600 bit/s (valore predefinito) ● il valore 4 (0x04) corrisponde a 19200 bit/s ● il valore 5 (0x05) corrisponde a 38400 bit/s ● il valore 6 (0x06) corrisponde a 57600 bit/s (può essere preso solamente per il modulo BMX NOM 0200) ● il valore 7 (0x07) corrisponde a 115200 bit/s (può essere preso solamente per il modulo BMX NOM 0200) Il byte 1 di questa parola corrisponde al formato: ● bit 8: numero di bit (1 = 8 bit (RTU), 0 = 7 bit (ASCII)) ● bit 9 = 1: gestione della parità (1 = con, 0 = senza) ● bit 10: tipo di parità (1 = dispari, 0 = pari) ● bit 11: numero di bit di stop (1 = 1 bit, 0 = 2 bit) ● bit 13: linea fisica (1 = RS232, 0 = RS485) ● bit 14: Segnali del modem DTR/DSR/DCD. Per il modulo BMX NOM 0200 e se viene selezionato RS232 questo bit può assumere 2 valori diversi: 1 significa che i segnali del modem vengono gestiti, 0 significa che non lo sono (valore predefinito per BMX P34 o se viene selezionato RS485) ● bit 15: Segnali di gestione del flusso hardware RTS/CTS. Se viene selezionato RS232 questo bit può assumere 2 valori diversi: 0 per RX/TX e 1 per RX/TX + RTS/CTS. Se viene selezionato RS485 il valore predefinito è 0 e corrisponde a RX/TX

Indirizzo	Tipo	Accesso	Significato
%KWr.m.c.2	INT	R	Valore immesso in ms di Stop con silenzio (dipende dalla velocità di trasmissione e dal formato selezionati). Il valore 0 significa che non è stato rilevato silenzio.
%KWr.m.c.3	INT	R	Questa parola corrisponde al tipo di polarizzazione: ● Il valore 0 su entrambi bit 14 e bit 15 non corrisponde a nessuna polarizzazione (Questo è il valore predefinito per BMX P34 o se viene selezionato RS232) ● bit 14: il valore 1 corrisponde a una polarizzazione a bassa impedenza (tipo Modbus) e può essere preso solamente per il modulo BMX NOM 0200 e se viene selezionato RS485 ● bit 15: il valore 1 corrisponde a una polarizzazione ad alta impedenza e può essere preso solamente per il modulo BMX NOM 0200 e se viene selezionato RS485
%KWr.m.c.5	INT	R	Questa parola corrisponde al tempo di ritardo RTS/CTS in centesimi di ms da 0 a 100 se è selezionato RS232. Il valore predefinito è 0 se è selezionato RS485.
%KWr.m.c.6	INT	R	Il bit 0 del Byte 0 può avere due valori: ● il valore 1 corrisponde alla casella di controllo Stop nell'area Stop alla ricezione per il carattere 1, quando selezionata ● il valore 0 corrisponde alla casella di controllo Stop nell'area Stop alla ricezione per il carattere 1, quando deselezionata Il bit 1 del Byte 0 può avere due valori: ● il valore 1 corrisponde alla casella di controllo Carattere incluso nell'area Stop alla ricezione per il carattere 1, quando selezionata ● il valore 0 corrisponde alla casella di controllo Carattere incluso nell'area Stop alla ricezione per il carattere 1, quando deselezionata Il byte 1 di questa parola corrisponde al valore immesso di stop alla ricezione del carattere da 0 a 255.
%KWr.m.c.7	INT	R	Il bit 0 del Byte 0 può avere due valori: ● il valore 1 corrisponde alla casella di controllo Stop nell'area Stop alla ricezione per il carattere 2, quando selezionata ● il valore 0 corrisponde alla casella di controllo Stop nell'area Stop alla ricezione per il carattere 2, quando deselezionata Il bit 1 del Byte 0 può avere due valori: ● il valore 1 corrisponde alla casella di controllo Carattere incluso nell'area Stop alla ricezione per il carattere 2, quando selezionata ● il valore 0 corrisponde alla casella di controllo Carattere incluso nell'area Stop alla ricezione per il carattere 2, quando deselezionata Il byte 1 di questa parola corrisponde al valore immesso di stop alla ricezione del carattere da 0 a 255.

Sezione 10.5
IODDT tipo T_GEN_MOD applicabili a tutti i moduli

Dettagli degli oggetti linguaggio dell'IODDT di tipo T_GEN_MOD

Introduzione

I moduli dei PLC Modicon M340 e M80 hanno un IODDT associato di tipo T_GEN_MOD.

Osservazioni

Generalmente, il significato dei bit viene fornito per lo stato 1 del bit. In casi specifici, si fornisce una spiegazione per ciascuno stato del bit.

Alcuni bit non vengono utilizzati.

Elenco di oggetti

La tabella seguente descrive gli oggetti dell'IODDT.

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
MOD_ERROR	BOOL	R	Bit di errore del modulo rilevato	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Parola di controllo di scambio del modulo	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Lettura in corso delle parole di stato del modulo	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Parola del rapporto di scambio	%MWr.m.MOD.1
STS_ERR	BOOL	R	Evento durante la lettura delle parole di stato del modulo	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Rilevamento della parola di errore interno del modulo	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	modulo inutilizzabile	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Canale(i) non operativo(i)	%MWr.m.MOD.2.1
BLK	BOOL	R	Morsettiera non cablata correttamente	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Anomalia di configurazione hardware o software	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Modulo non presente o non operativo	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Rilevamento parola di errore interno del modulo (solo estensione Fipio)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Errore interno, modulo non riparabile (solo estensione Fipio)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Canale(i) non operativo(i) (solo estensione Fipio)	%MWr.m.MOD.2.9

Simbolo standard	Tipo	Accesso	Significato	Indirizzo
BLK_EXT	BOOL	R	Morsettiera non cablata correttamente (solo estensione Fipio)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Anomalia di configurazione hardware o software (solo estensione Fipio)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Modulo non presente o non operativo (solo estensione Fipio)	%MWr.m.MOD.2.14

Sezione 10.6
Oggetti linguaggio e DDT dei dispositivi associati alla comunicazione Modbus

Nomi del DDT del dispositivo di comunicazione

Introduzione

Questa sezione descrive il **DDT del dispositivo di comunicazione** di Unity Pro.

Il nome DDT predefinito del dispositivo contiene le seguenti informazioni:
- ingresso e/o uscita modulo (simbolo ***X***)
- numero di inserimento del modulo (simbolo **#**)

Esempio: `MOD_COM_`***x***`_#`

Il tipo DDT predefinito del dispositivo contiene le seguenti informazioni:
- piattaforma con:
  - M per Modicon M340
- tipo di dispositivo (COM per digitale)
- funzione (NOM per BMX NOM 0200)
- direzione:
  - IN
  - OUT

Elenco dei DDT impliciti del dispositivo

La tabella seguente mostra l'elenco dei dispositivi Modicon M340 con il corrispondente nome e tipo DDT del dispositivo:

Nome DDT dispositivo	Tipo DDT dispositivo	Dispositivi Modicon M340
MOD_NOM_**#**	`T_M_COM_NOM`	BMX NOM 0200.3

Descrizione dei DDT impliciti del dispositivo

La tabella seguente mostra i bit della parola di stato T_M_COM_NOM:

Simbolo standard	Tipo	Significato	Accesso
MOD_HEALTH	BOOL	0 = il modulo ha rilevato un errore	lettura
		1 = il modulo funziona correttamente	
MOD_FLT	BYTE	rilevati byte di errore interni del modulo	lettura
COM_CH	ARRAY [0...1] di T_M_COM_NOM_CH	array di struttura	

La tabella seguente mostra i bit della parola di stato T_M_COM_NOM_CH[0...1]:

Simbolo standard		Tipo	Bit	Significato	Accesso
FCT_TYPE		WORD		0 = il canale non è utilizzato	lettura
				1 = il canale è utilizzato	
				3 = Modalità caratteri	
				7 = Slave MODBUS	
CH_HEALTH		BOOL		0 = il canale ha rilevato un errore	lettura
				1 = il canale funziona correttamente	
INPUT_SIGNALS [INT]	DCD	BOOL	0	La portante dati ha rilevato il segnale RS232 (applicabile solo al modulo BMX NOM 0200)	lettura
	CTS	BOOL	2	segnale RS232 Clear to send	lettura
	DSR	BOOL	3	Data set ready ha rilevato il segnale RS232 (applicabile solo al modulo BMX NOM 0200)	lettura
COMMAND [INT]	STOP_EXCH	BOOL	0	fronte di salita a 1: tutti gli scambi in corso sono interrotti.	lettura/scrittura

Descrizione delle istanze DDT del dispositivo esplicite

Scambi espliciti (lettura stato) - applicabili solo a canali di I/O Modicon M340 - gestiti con l'istanza EFB READ_STS_QX.

- L'indirizzo del canale di destinazione (ADDR) può essere gestito con ADDMX *(vedi Unity Pro, Comunicazione, Libreria dei blocchi funzione)* EF (collegare ADDMX OUT a ADDR)
- Il parametro di uscita READ_STS_QX *(vedi Unity Pro, Gestione I/O, Libreria dei blocchi funzione)* (STATUS) può essere collegato a un'istanza DDT "T_M_xxx_yyy_CH_STS" (variabile da creare manualmente), dove:
  - xxx rappresenta il tipo di dispositivo
  - yyy rappresenta la funzione

  Esempio: T_M_COM_NOM_CH_STS

La tabella seguente mostra i bit della parola di stato T_M_COM_NOM_CH_STS:

Tipo	Tipo	Accesso
STRUCT	T_M_COM_NOM_CH_STS	

La tabella seguente mostra i bit della parola di stato della struttura T_M_COM_NOM_CH_STS:

Simbolo standard		Tipo	Bit	Significato	Accesso
CH_FLT [INT]	NO_DEVICE	BOOL	0	nessun dispositivo operativo sul canale	lettura
	ONE_DEVICE_FLT	BOOL	1	dispositivo non utilizzabile sul canale	lettura
	BLK	BOOL	2	rilevata morsettiera guasta (non collegata)	lettura
	TO_ERR	BOOL	3	rilevato errore di timeout (cablaggio difettoso)	lettura
	INTERNAL_FLT	BOOL	4	rilevato errore interno o test automatico del canale	lettura
	CONF_FLT	BOOL	5	errore di configurazione rilevato: configurazioni hardware e software diverse	lettura
	COM_FLT	BOOL	6	problema di comunicazione con il PLC	lettura
	APPLI_FLT	BOOL	7	rilevato errore dell'applicazione (errore di regolazione o configurazione	lettura
PROTOCOL		BYTE		6 per master Modbus, 3 per modalità carattere	lettura
ADDRESS		BYTE		indirizzo slave	lettura

Capitolo 11
Cambio dinamico di protocollo

Argomento della sezione

Questo capitolo fornisce un'introduzione sul passaggio dinamico tra i protocolli Modbus e in modalità carattere.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sottosezioni:

Argomento	Pagina
Modifica del protocollo con i processori BMX P34 1000/2000/2010/20102/2020	244
Modifica del protocollo con il modulo BMX NOM 0200	246

Modifica del protocollo con i processori BMX P34 1000/2000/2010/20102/2020

Generale

Questa sezione descrive come cambiare il protocollo utilizzato da una comunicazione seriale CPU tramite il comando `WRITE_CMD(IODDT_VAR1)`. Questo comando può essere utilizzato per passare tra i seguenti tre protocolli:

- Slave Modbus
- Master Modbus
- Modalità carattere

NOTA: la variabile IODDT_VAR1 deve essere di tipo T_COM_MB_BMX.

Cambio di protocollo: principio

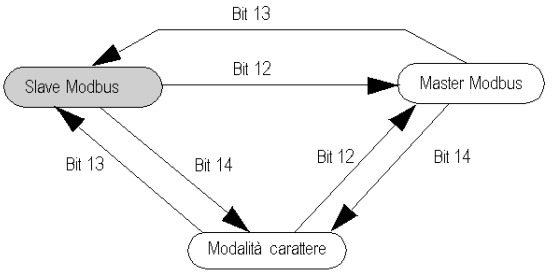
Occorre creare prima una variabile IODDT collegata al canale seriale del processore, quindi impostare a 1 il bit parola IODDT_VAR1.CONTROL (`%MWr.m.c.24`) che corrisponde al cambio di protocollo desiderato:

- `TO_MODBUS_MASTER` (Bit 12): il protocollo corrente viene cambiato in Modbus master
- `TO_MODBUS_SLAVE` (Bit 13): il protocollo corrente viene cambiato in Modbus slave
- `TO_CHAR_MODE` (Bit 14): il protocollo corrente viene cambiato in modalità caratteri.

NOTA: IODDT_VAR1.CONTROL (`%MWr.m.c.24`) fa parte della variabile IODDT IODDT_VAR1.

Successivamente, applicare l'istruzione `WRITE_CMD` alla variabile IODDT collegata al canale seriale processore.

Lo schema seguente mostra i cambi di protocollo da effettuare in base ai bit della parola IODDT_VAR1.CONTROL (`%MWr.m.c.24`) impostati a 1:



NOTA: per effettuare il passaggio da un protocollo a un altro, il processore deve essere inizialmente configurato in modalità Modbus slave.

Usi

Vengono utilizzati tre cambi di protocollo:

- Trasferimento a Modbus Master: il cambio di protocollo è un processo in due fasi:
  - passaggio dalla configurazione Modbus Slave alla configurazione Modbus Master,
  - ritorno alla configurazione iniziale Modbus Slave.

  Lo scopo della configurazione Modbus Master è l'invio di informazioni su un evento a un altro PLC. Quando si esegue un passaggio da una configurazione Modbus Slave a una Modbus Master, i parametri di trasmissione, segnale e linea fisica rimangono uguali. Cambiano solo i valori dei parametri seguenti, specifici della configurazione Modbus Master:
  - il Ritardo tra frame viene impostato al valore predefinito, che dipende dalla velocità di trasmissione
  - il ritardo alla risposta è impostato a 3.000ms
  - il numero di tentativi viene impostato a 3.
- Passaggio alla modalità caratteri: il cambio di protocollo è un processo in due fasi:
  - passaggio dalla configurazione Modbus Slave alla configurazione in modalità caratteri,
  - ritorno alla configurazione iniziale Modbus Slave.

  Lo scopo della configurazione in modalità caratteri è la comunicazione con un protocollo privato (un modem, ad esempio). Quando si esegue un passaggio da una configurazione Modbus Slave a una in modalità caratteri, i parametri di trasmissione, segnale e linea fisica rimangono uguali. Solo il parametro di fine messaggio specifico della modalità carattere viene impostato su Stop con silenzio e con un timeout di 1000ms.
- Passaggio ai protocolli in modalità caratteri e Modbus master. Il cambio di protocollo è un processo in tre fasi:
  - passaggio dalla configurazione Modbus Slave alla configurazione in modalità caratteri,
  - passaggio dalla configurazione in modalità caratteri alla configurazione Modbus master,
  - ritorno alla configurazione iniziale Modbus Slave.

  Lo scopo della configurazione in modalità caratteri è la comunicazione con un protocollo privato (un modem, ad esempio). Al termine dello scambio, l'utente passa alla configurazione Modbus Master per inviare informazioni su un evento a un altro PLC. Dopo l'invio del messaggio, l'utente ritorna alla configurazione iniziale Modbus Slave.

NOTA: Per tutti i tre casi, la configurazione predefinita rimane Modbus Slave.

Avvii a caldo e a freddo

I cambi di protocollo non sono influenzati dai bit `%S0` e `%S1` (i bit impostati a 1 durante un avvio a freddo e a caldo, rispettivamente). Tuttavia, un avvio a freddo o a caldo del PLC configura la porta seriale ai valori predefiniti o ai valori programmati nell'applicazione.

Modifica del protocollo con il modulo BMX NOM 0200

Generale

Questa sezione descrive come cambiare il protocollo utilizzato da una comunicazione seriale BMX NOM 0200 tramite il comando `WRITE_CMD(IODDT_VAR1)` .

Questo comando può essere utilizzato per passare tra i seguenti tre protocolli:

- Slave Modbus
- Master Modbus
- Modalità carattere

NOTA: La variabile IODDT_VAR1 deve essere di tipo T_COM_MB_BMX o T_COM_MB_BMX CONF EXT.

Cambio di protocollo: principio

Occorre creare prima una variabile IODDT collegata al canale seriale del processore, quindi impostare a 1 il bit parola IODDT_VAR1.CONTROL (`%MWr.m.c.24`) che corrisponde al cambio di protocollo desiderato:

- `TO_MODBUS_MASTER` (Bit 12): il protocollo corrente viene cambiato in Modbus master
- `TO_MODBUS_SLAVE` (Bit 13): il protocollo corrente viene cambiato in Modbus slave
- `TO_CHAR_MODE` (Bit 14): il protocollo corrente viene cambiato in modalità caratteri.

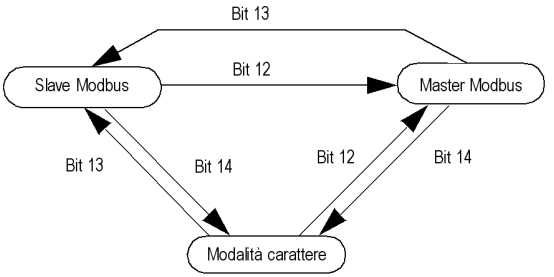
NOTA: Un singolo bit può essere impostato su 1 alla volta. L'impostazione di diversi bit su 1 risulterà in un errore.

NOTA: IODDT_VAR1.CONTROL (`%MWr.m.c.24`) fa parte della variabile IODDT IODDT_VAR1.

Successivamente, applicare l'istruzione `WRITE_CMD` alla variabile IODDT collegata al canale seriale processore.

NOTA: Fare attenzione che i due master (sullo stesso bus) non inviino richieste simultaneamente, altrimenti le richieste vanno perdute e ogni report avrà un risultato errato che potrebbe essere 16#0100 (impossibile elaborare la richiesta) o 16#ODFF (slave non presente).

Lo schema seguente mostra i cambi di protocollo da effettuare in base ai bit della parola IODDT_VAR1.CONTROL (`%MWr.m.c.24`) impostati a 1:



Usi

Vengono utilizzati tre cambi di protocollo:

- Trasferimento dal Modbus Slave al Modbus Master:
  Lo scopo della configurazione Modbus Master è l'invio di informazioni su un evento a un altro PLC. Quando si esegue un passaggio da una configurazione Modbus Slave a una Modbus Master, i parametri di trasmissione, segnale e linea fisica rimangono uguali. Cambiano solo i valori dei parametri seguenti, specifici della configurazione Modbus Master:
  - il Ritardo tra frame viene impostato al valore predefinito, che dipende dalla velocità di trasmissione
  - Il ritardo alla risposta è impostato a 3s
  - il numero di tentativi viene impostato a 0.
- passaggio dalla configurazione Modbus Slave/Master alla configurazione in modalità caratteri
  Il passaggio alla modalità caratteri viene utilizzato per inviare i comandi AT a un modem. Quando si esegue un passaggio da una configurazione Modbus a una in modalità caratteri, i parametri di trasmissione, segnale e linea fisica rimangono uguali. Solo il parametro di rilevamento fine messaggio specifico della modalità carattere viene impostato su stop alla ricezione del carattere di fine `x0d` .
- Trasferimento dalla Modalità Caratteri a Modbus Master e Modbus Slave:
  Lo scopo della configurazione in modalità caratteri è la comunicazione con un protocollo privato (un modem, ad esempio). Al termine dello scambio, l'utente passa alla configurazione Modbus Master (con il ritardo della risposta impostato su 3s e il numero di tentativi impostato su 0) per inviare informazioni su un evento a un altro PLC. Dopo l'invio del messaggio, l'utente ritorna alla configurazione Modbus Slave: il numero dello slave viene impostato sul valore archiviato nella memoria FLASH o su 248 se non ne è presente nessuno.

Avvii a caldo e a freddo

I cambi di protocollo non sono influenzati dai bit `%S0` e `%S1` (i bit impostati a 1 durante un avvio a freddo e a caldo, rispettivamente). Tuttavia, un avvio a freddo o a caldo del PLC configura la porta seriale ai valori predefiniti o ai valori programmati nell'applicazione.

NOTA: la configurazione predefinita del modulo è la seguente: per essere facilmente configurabile da un computer come un PC, il canale 0 è configurato in modalità slave RS232, e il canale 1 in modalità RS485. Altri parametri sono: 19200 baud, RTU, pari, 1 bit di arresto, nessun controllo di flusso, 1,75ms come ritardo di frame predefinito, numero dello slave 248.

Parte IV
Avvio rapido: Esempio dell’implementazione del collegamento seriale

Panoramica

Questa sezione illustra un esempio di implementazione del collegamento seriale.

Contenuto di questa parte

Questa parte contiene i seguenti capitoli:

Capitolo	Titolo del capitolo	Pagina
12	Descrizione dell’applicazione	251
13	Installazione dell’applicazione con Unity Pro	253
14	Avvio dell’applicazione	281

Capitolo 12
Descrizione dell’applicazione

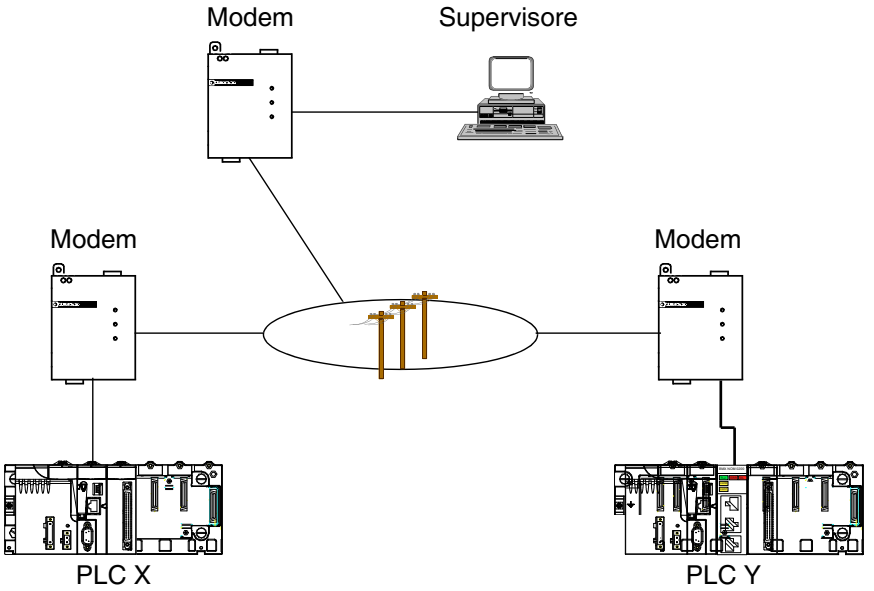
Panoramica dell’applicazione

In breve

L’applicazione descritta in questo documento è un’applicazione di comunicazione Modbus tramite modem.

Esempio

Nella seguente figura è illustrato un esempio:



I dispositivi comunicano tra di loro per mezzo di modem. Il supervisore è il master Modbus, mentre i PLC X e Y sono gli slave.

Nell’esempio illustrato si intendono scrivere i valori dell’area dati del PLC X nel PLC Y.

A questo scopo, PLC X deve diventare Modbus Master.

Ogni giorno, il supervisore comunica con i PLC per ottenere informazioni.

Se si verifica un allarme sul PLC X, questo passa in modalità Modbus Master e invia i dati al PLC Y.

Per semplificare la programmazione, i modem sono stati inizializzati con i parametri corretti tramite un terminale di programmazione. Questi parametri sono archiviati nella memoria non volatile dai comandi AT&W.

Modalità di funzionamento

L’applicazione funziona come segue:

Passo	Azione
1	La porta del PLC X passa alla modalità carattere.
2	Il PLC X invia un messaggio di composizione al modem.
3	La porta del PLC X passa alla modalità Modbus Master.
4	Il PLC master (X) invia i dati al PLC slave (Y).
5	La porta passa in modalità carattere.
6	Il PLC X invia un messaggio di disconnessione al modem.
7	La porta del PLC X passa alla modalità Modbus Slave.

Capitolo 13
Installazione dell’applicazione con Unity Pro

Oggetto del capitolo

Questo capitolo descrive la procedura per la creazione dell’applicazione illustrata. Indica, in generale e in dettaglio, i vari passi da eseguire per creare i diversi componenti dell’applicazione.

Contenuto di questo capitolo

Questo capitolo contiene le seguenti sezioni:

Sezione	Argomento	Pagina
13.1	Presentazione della soluzione utilizzata	254
13.2	Sviluppo dell’applicazione	255

Sezione 13.1
Presentazione della soluzione utilizzata

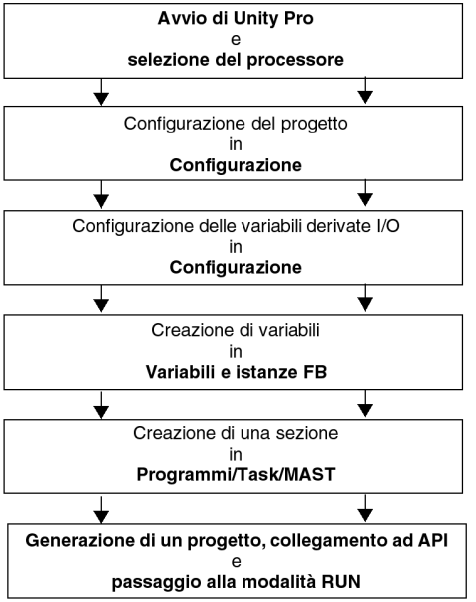
Passi della procedura che utilizza Unity Pro

In breve

Il seguente schema logico descrive i vari passi da seguire per creare l’applicazione. È necessario rispettare un ordine cronologico per definire correttamente tutti gli elementi dell’applicazione.

Descrizione

Descrizione dei diversi tipi:



Sezione 13.2
Sviluppo dell’applicazione

Argomento della sezione

Questa sezione descrive la procedura di creazione dell’applicazione mediante Unity Pro.

Contenuto di questa sezione

Questa sezione contiene le seguenti sottosezioni:

Argomento	Pagina
Creazione di un progetto	256
Dichiarazione delle variabili	261
Uso del modem	265
Procedura per la programmazione	267
Struttura della programmazione	268
Programmazione	271

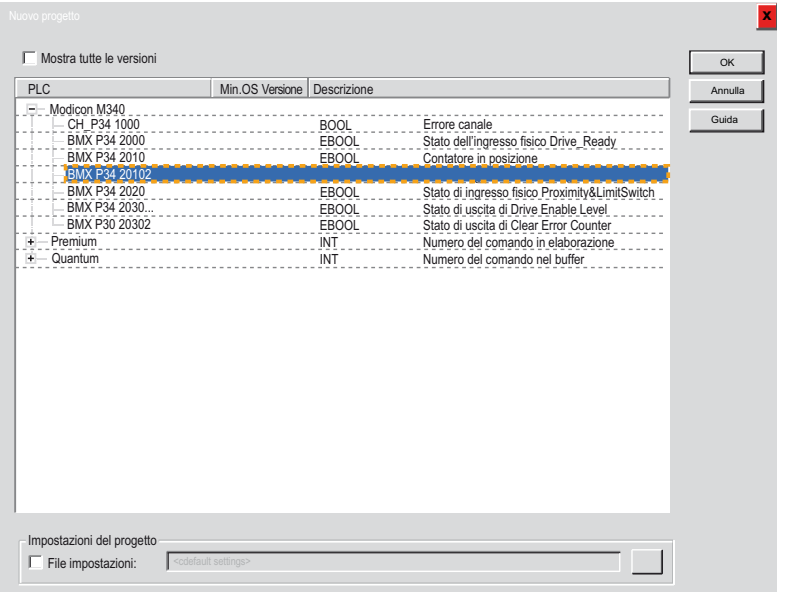
Creazione di un progetto

In breve

Per procedere allo sviluppo dell’esempio, è necessario creare un progetto principale con il PLC X per la configurazione del PLC X e la dichiarazione di tutte le variabili necessarie e la programmazione dell’applicazione. Inoltre, è necessario creare un progetto separato per la configurazione del PLC Y.

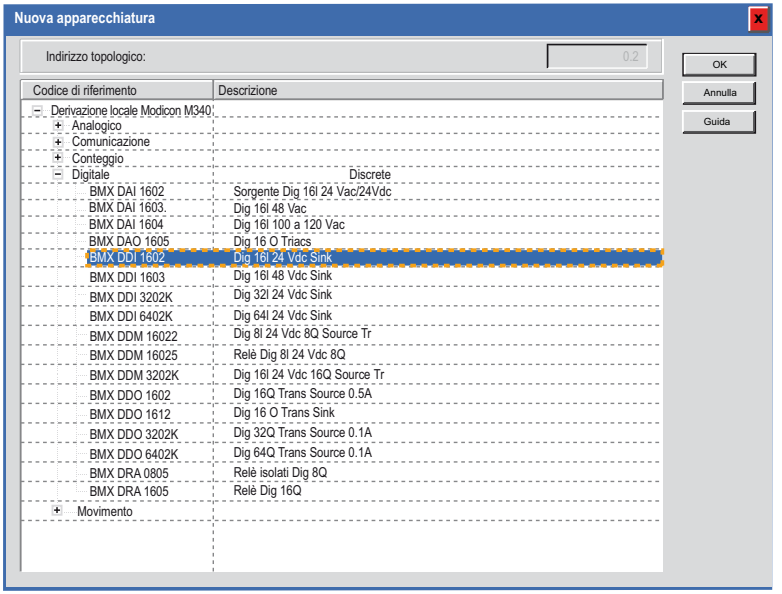
Procedura per la creazione di un progetto

La seguente tabella illustra la procedura di creazione di un progetto mediante Unity Pro.

Passo	Azione
1	Avviare il software Unity Pro.
2	Fare clic su File, Nuovo per selezionare un processore BMX P34 20102: 
3	Confermare scegliendo OK.

Selezione moduli di ingresso digitali

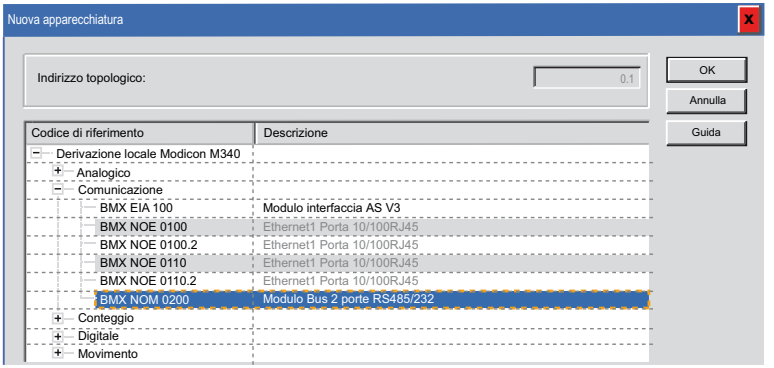
Nella seguente tabella è descritta la procedura per selezionare il modulo digitale richiesto dal PLC X.

Passo	Azione
1	In `Browser di progetto` fare doppio clic su `Configurazione`, quindi su `0:PLC Bus` e su `0:BMX XBP •••` (dove 0 è il numero di rack),
2	Nella finestra `PLC Bus`, fare doppio clic su uno slot (ad esempio il numero 1) per selezionarlo,
3	Scegliere il modulo di ingresso di conteggio `BMX DDI 1602` situato nell'elenco moduli `Digitali` , 
4	Confermare scegliendo OK.

Selezione Modulo BMX NOM 0200

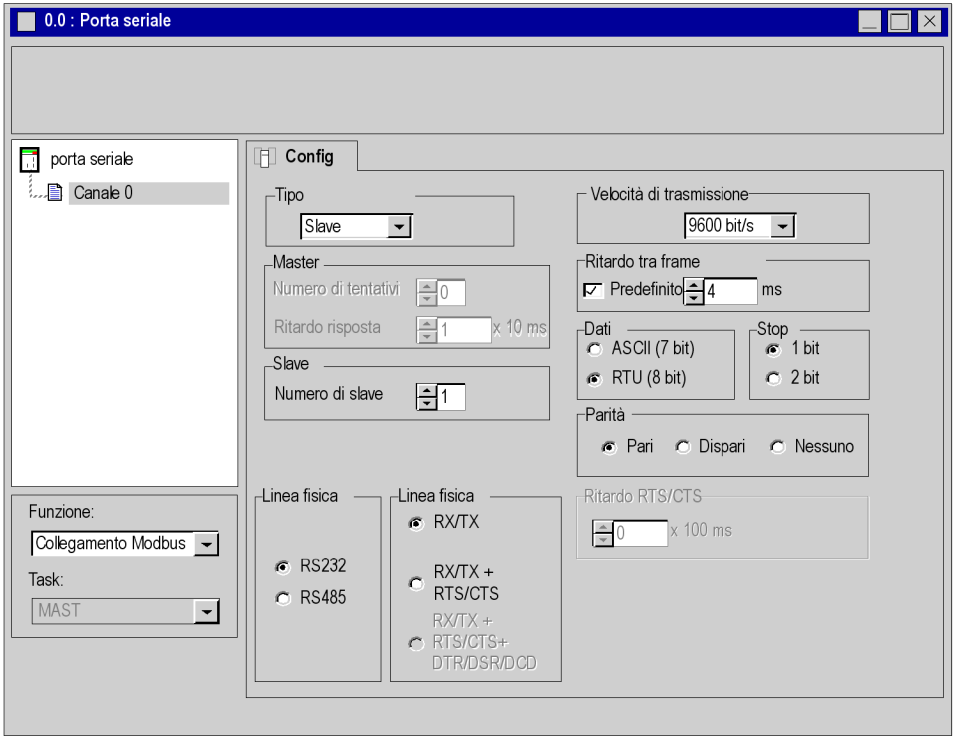
In questo esempio, viene utilizzato un modulo BMX NOM 0200 nel PLC Y per il collegamento seriale con il modem. Di conseguenza deve essere aggiunto al progetto associato con il PLC Y.

La tabella seguente mostra la procedura per selezionare il modulo BMX NOM 0200.

Passo	Azione
1	In `Browser di progetto` fare doppio clic su `Configurazione`, quindi su `0:PLC Bus` e su `0:BMX XBP •••` (dove 0 è il numero di rack),
2	Nella finestra `PLC Bus`, fare doppio clic su uno slot (ad esempio il numero 1) per selezionarlo,
3	Scegliere il modulo di comunicazione `BMX NOM 0200` situato nell’elenco moduli `Comunicazione` , 
4	Confermare scegliendo OK.

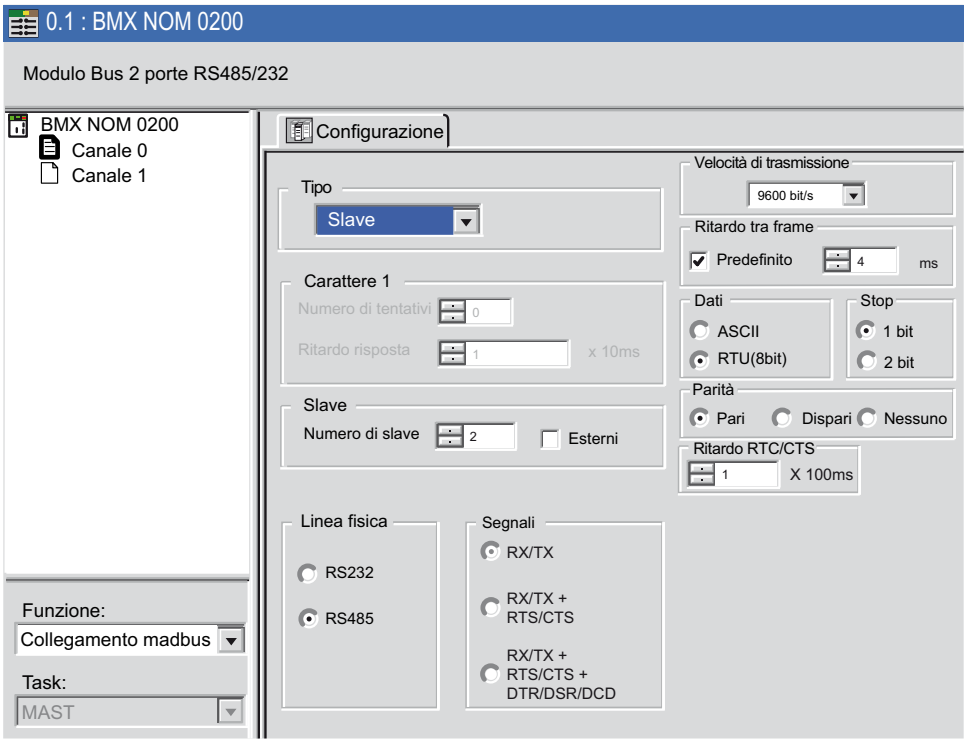
Configurazione della porta seriale del processore

Nella seguente tabella è descritta la procedura di configurazione della porta seriale del processore PLC X come slave Modbus:

Passo	Azione
1	Nel `Browser del progetto`, fare doppio clic su `Configurazione`, quindi `0: BMX XBP 0800` e infine `0: BMX P34 20102`. Quindi fare doppio clic su `Porta seriale` per accedere alla finestra `0.0:Porta seriale`. 
2	Selezionare il tipo `Slave`.
3	Selezionare `9600 bit/s` per la velocità di trasmissione.
4	Selezionare `RS232` per la linea fisica.
5	Selezionare `RTU (8bit)` per il tipo di dati.
6	Chiudere la finestra e confermare con OK.

Configurazione canale seriale BMX NOM 0200

Nella seguente tabella è descritta la procedura di configurazione del canale seriale del modulo PLC Y BMX NOM 0200 come slave Modbus:

Passo	Azione
1	In `Browser di progetto` fare doppio clic su `Configurazione` quindi su `0:BMX XBP 0800` e su `0:BMX NOM 0200` per accedere alla finestra `0.x:BMX NOM 0200` (dove x è il numero dello slot, per esempio x=1). 
2	Selezionare il `Canale 0`.
3	Selezionare il `collegamento Modbus` per la funzione.
4	Selezionare il tipo `Slave`.
5	Selezionare `9600 bit/s` per la velocità di trasmissione.
6	Selezionare `RS232` per la linea fisica.
7	Selezionare `RX/TX + RTS/CTS + DTR/DSR/DCD` per i segnali.
8	Selezionare `100 ms` per il ritardo RTS/CTS.
9	Selezionare `RTU (8bit)` per il tipo di dati.
10	Chiudere la finestra e confermare con OK.

Dichiarazione delle variabili

In breve

Tutte le variabili utilizzate nelle varie sezioni del programma devono essere dichiarate.

Le variabili non dichiarate non possono essere utilizzate nel programma.

NOTA: Per ulteriori informazioni, vedere la guida in linea di Unity Pro (fare clic su ?, su `Unity`, su `Unity Pro Software`, su `Modalità operative`, e quindi su `Editor dati`).

Procedura per la dichiarazione delle variabili

La tabella seguente mostra la procedura per la dichiarazione delle variabili dell'applicazione:

Passo	Azione
1	In `Browser del progetto / Variabili e istanze FB`, fare doppio clic su `Variabili elementari`
2	Nella finestra `Editor dati`, selezionare la casella nella colonna Nome ed immettere un nome per la prima variabile.
3	Selezionare quindi un Tipo di variabile.
4	Una volta dichiarate tutte le variabili, è possibile chiudere la finestra.

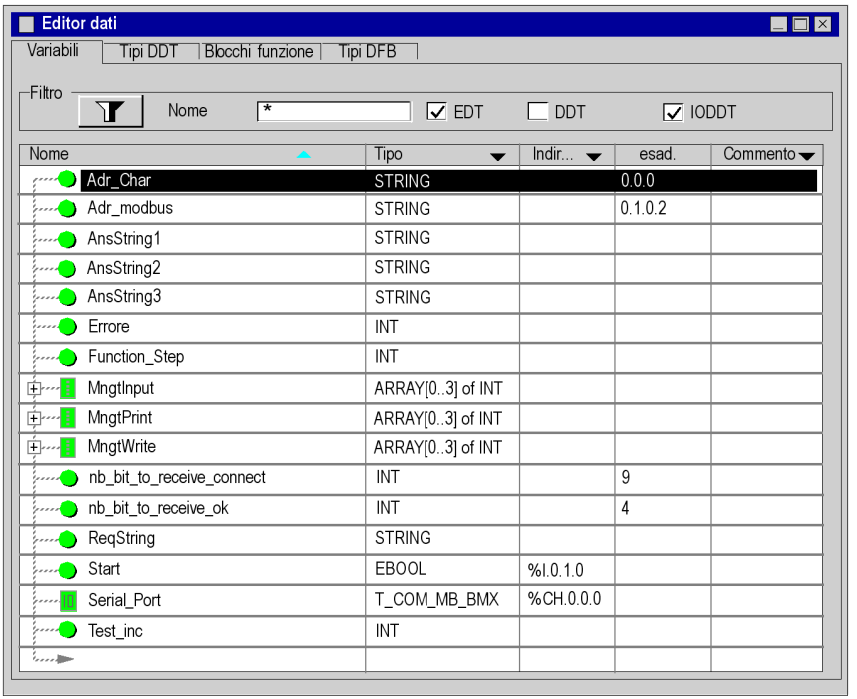
Variabili utilizzate per l'applicazione

La tabella seguente riporta i dettagli delle variabili utilizzate nell'applicazione e dichiarate nel progetto associato al PLC X:

Variabile	Tipo	Definizione
Adr_Char	STRING	Indirizzo della porta seriale del PLC master.
Adr_modbus	STRING	Indirizzo del canale seriale Modbus Slave PLC (canale 0 del modulo BMX NOM 0200).
AnsString1	STRING	Prima stringa caratteri di risposta modem.
AnsString2	STRING	Seconda stringa caratteri di risposta modem.
AnsString3	STRING	Terza stringa caratteri di risposta modem.
Errore	INT	Codice errore funzione.
Function_Step	INT	Passo funzione.
MngtInput	ARRAY[0..3] of INT	Array dei parametri di comunicazione per il blocco INPUT_CHAR.
MngtPrint	ARRAY[0..3] of INT	Array dei parametri di comunicazione per il blocco PRINT_CHAR.

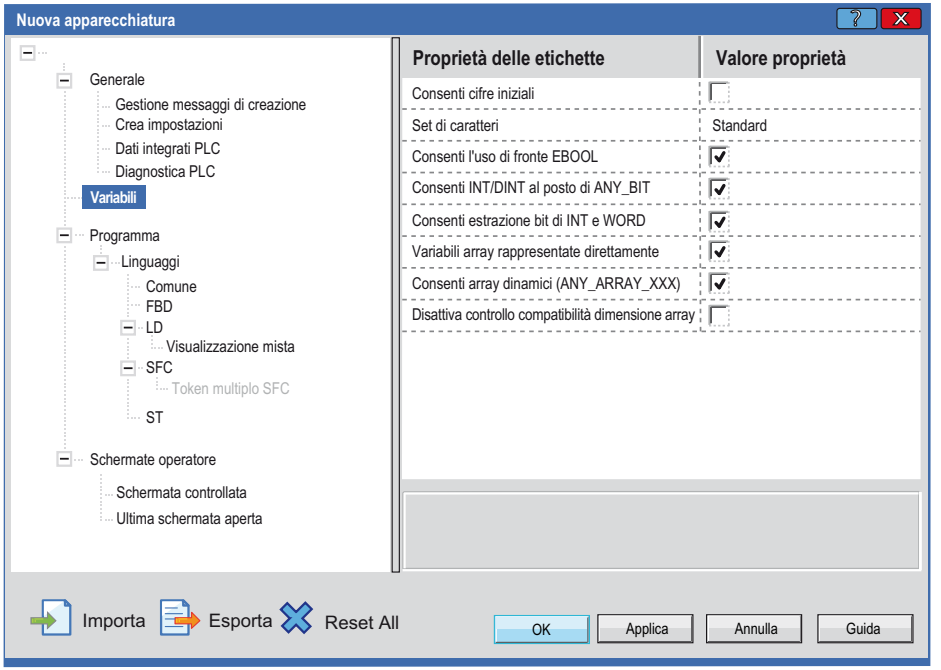
Variabile	Tipo	Definizione
MngtWrite	ARRAY[0..3] of INT	Array dei parametri di comunicazione per il blocco WRITE_VAR.
n_caratt_da_ricevere_collegato	INT	Numero di caratteri da ricevere: connessione modem
n_caratt_da_ricevere_ok	INT	Numero di caratteri da ricevere: messaggio di conferma modem
ReqString	STRING	Risposta modem.
Start	EBOOL	Modalità di avvio (segnale proveniente dal canale 0 del modulo BMX DDI 1602).
Serial_Port	T_COM_MB_BMX	Oggetto di I/O della porta seriale
Test_inc	INT	Valore di incremento

La schermata seguente mostra le variabili dell'applicazione create utilizzando l'editor dati:

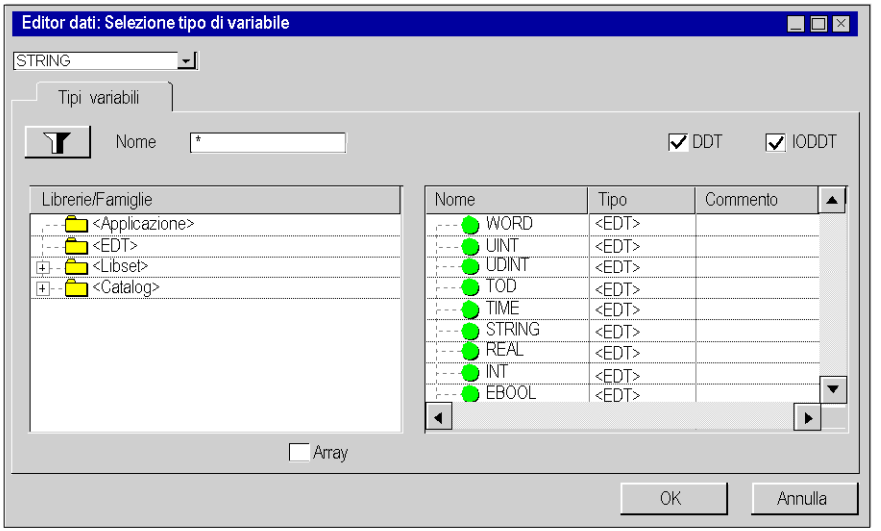


Dichiarazione di un tipo di array

Prima di dichiarare un tipo di array, fare clic su **Strumenti/Impostazioni progetto/Variabili** quindi selezionare "Variabili array rappresentate direttamente" e "Consenti array dinamici"



Nella seguente tabella è indicata la procedura per dichiarare un tipo di array:

Passo	Azione
1	Nel `Browser del progetto`, fare clic su `Variabili e istanze FB`.
2	Fare clic sulla colonna `Nome` e specificare un nome per la variabile.
3	Fare doppio clic sulla colonna `Tipo`, quindi fare clic sul pulsante ... . Viene visualizzata la finestra `Selezione tipo di variabile`: 
4	Selezionare il tipo di variabile desiderato (ad esempio, fare clic su `<EDT>` e selezionare `INT`), quindi fare clic sulla casella di controllo Array.
5	Modificare l'intervallo, quindi confermare con `OK`.

Dichiarazione degli oggetti di I/O

Per la dichiarazione delle variabili derivate I/O, aprire la finestra `Selezione tipo variabili` come descritto nella procedura di sopra e fare clic su `<Catalogo>` per accedere alle variabili di tipo `<IODDT>` (selezionare `T COM MB BMX` per esempio), quindi confermare con `OK`.

Uso del modem

Descrizione

Per interfacciare i modem telefonici ai PLC, è necessario conoscere tre comandi. Si tratta dei seguenti comandi:

- Inizializza modem
- Rinumera
- Scollega modem.

Prima di inviare un messaggio ASCII o Modbus, è necessario inviare al modem un messaggio di inizializzazione seguito da un messaggio di composizione.

Una volta che la connessione tra i due modem funziona regolarmente, è possibile inviare un numero illimitato di messaggi ASCII o Modbus.

Dopo aver inviato tutti i messaggi, occorre inviare al modem la stringa di disconnessione.

Inizializzazione del modem

I due modem devono essere configurati con le stesse caratteristiche della porta seriale:

- velocità dati: 9600 baud
- frame caratteri: 8 bit / parità pari / 1 bit stop
- modulazione di linea: V32.

Quindi definire "+" come carattere di escape (comando: ATS2=43).

Esempio di comando di inizializzazione:

ATQ0&Q0E0&K0V1

con:

- Q0: abilita il codice risultato
- &Q0: DTR è sempre assunto (ON)
- E0: disabilita i caratteri di eco
- &K0: nessun controllo di flusso
- V1: codici risultato parola.

Composizione del modem

Il messaggio di composizione permette di inviare al modem un numero di telefono.

Nel messaggio devono essere utilizzati solo i comandi AT relativi alla composizione.

Esempio:

- **Composizione a frequenza:** ATDT6800326<CR><LR>
- **Composizione a impulsi:** ATDP6800326<CR><LF>
- **Composizione a frequenza con attesa del segnale:** ATDTW6800326<CR><LF>

Disconnessione del modem

Riportare il modem alla modalità comando inviando tre volte il carattere di escape.

A questo punto è possibile inviare il comando di disconnessione "ATH0".

Sequenza escape: "+++" (codice modem: OK)

Comando di scollegamento: "ATH0" (codice modem: OK).

Procedura per la programmazione

Procedura da seguire

Nella tabella è descritta la procedura per programmare l'applicazione.

Passo	Azione	Dettagli
1	Preparazione della porta di comunicazione.	• Passare dalla modalità Modbus Slave alla modalità Carattere inviando un comando WRITE_CMD *(vedi pagina 267)* alla porta seriale. • Per una trasmissione via modem, inviare il comando HAYES utilizzando il blocco PRINT_CHAR per configurare il modem *(vedi pagina 265)*. • Per una trasmissione via modem, inviare il comando HAYES utilizzando il blocco PRINT_CHAR. Il messaggio di composizione permette di inviare al modem un numero di telefono *(vedi pagina 265)*.
2	Modalità Modbus Master	• Passare alla modalità Modbus Master con il comando WRITE_CMD *(vedi pagina 267)* . • Inviare i dati da scrivere al PLC slave.
3	Reset della porta di comunicazione.	• Passare alla modalità carattere con il comando WRITE_CMD *(vedi pagina 267)* . • Per una trasmissione via modem, inviare il carattere di escape, quindi il comando scollega per inviare un messaggio di disconnessione al modem *(vedi pagina 266)* con il blocco PRINT_CHAR. • Ritornare alla modalità iniziale della porta seriale (Modbus Slave) con il comando WRITE_CMD *(vedi pagina 267)* .

Scrittura delle parole di comando

Per inviare un comando WRITE_CMD a una porta di comunicazione seguire la procedura descritta:

Passo	Azione	Dettagli
1	Verificare che non vi siano comandi in attesa.	Prima di eseguire un comando WRITE_CMD, verificare che non vi siano scambi in corso con l'oggetto di linguaggio `EXCH_STS` (%MWr.m.c.0). Per aggiornare questa parola, usare il blocco READ_STS.
2	Assegnare la parola di comando.	Per poter eseguire il comando richiesto, a questo punto è necessario modificare il valore dell'oggetto linguaggio di comando. Per un collegamento Modbus, l'oggetto linguaggio è la parola interna `CONTROL` (%MWr.m.c.24). Ad esempio, per passare dalla modalità Modbus alla modalità carattere, il bit 14 della parola %MWr.m.c.24 è impostato a 1. **Nota:** Prima di trasmettere il comando WRITE_CMD è necessario far passare un solo bit di comando da 0 a 1.
3	Invio del comando	Infine, deve essere eseguito un comando WRITE_CMD per riconoscere il comando.

Struttura della programmazione

Commenti ai passi

Numero del passo	Descrizione del passo	Elemento
0	Stato iniziale della funzione Quando il bit di avvio passa a 1, inizializzare l'errore su 0 e procedere al passo 5.	Modem
5	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando. Passare alla modalità carattere e inizializzare il contatore Test_inc su 0. Passare al punto 10.	

10	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando.
	Azzerare il bit di comando TO_CHAR_MODE.
	● Se non vi è un errore sulla porta seriale
	  ● e la modalità caratteri è attiva, passare al punto 15.
	  ● e la modalità caratteri è attiva, incrementare Test_inc e ritentare il passo 10 fino a 1000 volte. Dopo 1000 tentativi falliti, impostare l'Errore su 10 e procedere al passo 130.
	● Se vi è un errore sulla porta seriale,
	  ● impostare l'Errore su 10.
	  ● Passare al punto 130.	

15	Inviare un comando di composizione al modem tramite il blocco PRINT_CHAR. Passare al punto 20.	
20	Se il risultato di PRINT_CHAR è risolutivo, passare al punto 25, altrimenti impostare l'Errore su 20 e passare al punto 130.	
25	Attesa per la risposta del modem tramite il blocco INPUT_CHAR. Una volta ricevuta la risposta di stringa, procedere al passo 30.	
30	Se il risultato di INPUT_CHAR è risolutivo, passare al punto 35, altrimenti impostare l'Errore su 30 e passare al punto 130.	
35	Se il modem risponde come previsto, passare al punto 40, altrimenti impostare l'Errore su 35 e procedere al punto 130.	
40	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando. Passare alla modalità Modbus Master e inizializzare il contatore Test_inc su 0. Passare al punto 45.	Modalità Modbus Master

45	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando.
	Azzerare il bit di comando TO_CHAR_MODE.
	● Se non vi è un errore sulla porta seriale
	  ● e la modalità caratteri è attiva, passare al punto 50.
	  ● e la modalità caratteri è attiva, incrementare Test_inc e ritentare il passo 45 fino a 1000 volte. Dopo 1000 tentativi falliti, impostare l'Errore su 45 e procedere al passo 130.
	● Se vi è un errore sulla porta seriale,
	  ● impostare l'Errore su 45.
	  ● Passare al punto 130.	

Numero del passo	Descrizione del passo	Elemento
50	Inizializzazione del parametro del blocco WRITE_VAR. Inviare dati di scrittura al PLC mediante la funzione WRITE_VAR. Passare al punto 55.	Funzione di scrittura
55	Se il risultato di WRITE_CHAR è risolutivo, passare al punto 60, altrimenti impostare l'Errore su 55 e passare al punto 130.	
60	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando. Passare alla modalità carattere e inizializzare il contatore Test_inc su 0. Passare al punto 65	Modalità caratteri

65	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando.
	Azzerare il bit di comando TO_CHAR_MODE.
	● Se non vi è un errore sulla porta seriale
	  ● e la modalità caratteri è attiva, passare al punto 70.
	  ● e la modalità caratteri è attiva, incrementare Test_inc e ritentare il passo 65 fino a 1000 volte. Dopo 1000 tentativi falliti, impostare l'Errore su 65 e procedere al passo 130.
	● Se vi è un errore sulla porta seriale,
	  ● impostare l'Errore su 65.
	  ● Passare al punto 130.	

70	Inviare una sequenza di escape al modem con il blocco PRINT_CHAR. Passare al punto 75.	Modem
75	Se il risultato di PRINT_CHAR è risolutivo, passare al punto 80, altrimenti impostare l'Errore su 75 e passare al punto 130.	
80	Attesa per la risposta del modem tramite il blocco INPUT_CHAR. Una volta ricevuta la risposta di stringa, procedere al passo 85.	
85	Se il risultato di INPUT_CHAR è risolutivo, passare al punto 90, altrimenti impostare l'Errore su 85 e passare al punto 130.	
90	Se il modem risponde come previsto, passare al punto 95, altrimenti impostare l'Errore su 90 e procedere al punto 130.	
95	Inviare un comando di disconnessione al modem mediante il blocco PRINT_CHAR. Passare al punto 100.	
100	Se il risultato di PRINT_CHAR è risolutivo, passare al punto 105, altrimenti impostare l'Errore su 100 e passare al punto 130.	
105	Attesa per la risposta del modem tramite il blocco INPUT_CHAR. Una volta ricevuta la risposta di stringa, procedere al passo 110.	
110	Se il risultato di INPUT_CHAR è risolutivo, passare al punto 115, altrimenti impostare l'Errore su 110 e passare al punto 130.	
115	Se il modem risponde come previsto, passare al punto 120, altrimenti impostare l'Errore su 115 e procedere al punto 130.	

Numero del passo	Descrizione del passo	Elemento
120	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando. Passare alla modalità Modbus Slave e inizializzare il contatore Test_inc su 0. Passare al punto 125.	Modalità Modbus Slave

125	Leggere lo stato della porta seriale e verificare che non sia attivo alcun comando.
	Azzerare il bit di comando TO_CHAR_MODE.
	● Se non vi è un errore sulla porta seriale
	  ● e la modalità caratteri è attiva, passare al punto 130.
	  ● e la modalità caratteri è attiva, incrementare Test_inc e ritentare il passo 125 fino a 1000 volte. Dopo 1000 tentativi falliti, impostare l’Errore su 125 e procedere al passo 130.
	● Se vi è un errore sulla porta seriale,
	  ● impostare l’Errore su 125.
	  ● Passare al punto 130.	

130	Ritornare al punto 0.	

Programmazione

Programmazione in linguaggio ST

L'esempio è programmato in linguaggio ST. La sezione dedicata si trova sotto lo stesso task master (MAST).

```
CASE Function_Step OF
0: (* Inizializzazione *)
 IF (Start) THEN (* flag attivatore *)
 Error := 0;
 Function_Step := 5; (* passo successivo *)
 END_IF;
5: (* Invia comando per commutare porta seriale dalla modalità Slave Modbus a Modalità caratteri
*)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* nessun comando attivo *)
 Serial_port.CONTROL := 16#00; (* reset parola di controllo *)
 (* imposta bit comando TO_CHAR_MODE *)
 SET(Serial_port.TO_CHAR_MODE);
 WRITE_CMD (Serial_port); (* invio comando *)
 Test_inc := 0; (* inizializza Riprova conteggio *)
 Function_Step := 10; (* passo successivo *)
 END_IF;
10: (* Risultato test del comando di commutazione a Modalità caratteri *)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* comando completato *)
 (* reset bit di comando TO_CHAR_MODE *)
 RESET(Serial_port.TO_CHAR_MODE);
 IF (Serial_port.EXCH_RPT = 0) THEN (* nessun errore *)
 IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)
 THEN (* Modalità caratteri OK *)
 Function_Step := 15; (* passo successivo *)
 ELSE
 Test_inc := Test_inc + 1;
 IF (Test_inc > 1000) THEN
```

```
 Error := 10; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
 ELSE (* errore d’invio comando alla porta *)
 Error := 10; (* errore *)
 Function_Step := 130;
 END_IF;
 END_IF;

15: (* Invia comando di composizione al modem *)
 (*I numeri di telefono devono essere inseriti tra ‘ATDT’ e ‘$N’*)
 ReqString := ’ATDT4001$N’; (* messaggio chiamata *)
 MngtPrint[2] := 500; (* timeout *)
 MngtPrint[9] := 9; (* dimensione scambio in byte *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
 Function_Step := 20;
20: (* Risultato funzione test PRINT_CHAR *)
 IF (NOT MngtPrint[0].0) THEN
 IF (MngtPrint[1] = 0) THEN
 Function_Step := 25; (* success : passo successivo *)
 ELSE
 Error := 20; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
25: (* Attesa risposta via INPUT_CHAR *)
 MngtInput[2] := 500; (* timeout *)
 AnsString1:=’ ’;
 (* attesa risposta modem *)
 INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_connect, MngtInput, AnsString1);
 Function_Step := 30; (* passo successivo *)
```

```
30: (* Risultato funzione test INPUT_CHAR *)
 IF (NOT MngtInput[0].0) THEN
 IF (MngtInput[1] = 0) THEN
 Function_Step := 35; (* success : passo successivo *)
 ELSE
 Error := 30; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;

35: (* Risposta test modem *)
 IF (AnsString1 = ’$NCONNET’) THEN
 Function_Step := 40; (* success : passo successivo *)
 ELSE
 Error := 35; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;

40: (* Invio comando di commutazione porta seriale dalla Modalità caratteri a Modbus Master *)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* nessun comando attivo *)
 Serial_port.CONTROL := 16#00; (* reset parola di controllo *)
 (* imposta bit di comando TO_MODBUS_MASTER *)
 SET(Serial_port.TO_MODBUS_MASTER);
 WRITE_CMD (Serial_port); (* invio comando *)
 Test_inc := 0; (* inizializza Riprova conteggio *)
 Function_Step := 45; (* passo successivo *)
 END_IF;

45: (* Risultato test del comando di commutazione a modalità Modbus Master *)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* comando completato *)
 (* bit di comando TO_MODBUS_MASTER *)
```

```
RESET(Serial_port.TO_MODBUS_MASTER);
IF (Serial_port.EXCH_RPT = 0) THEN (* nessun errore *)
IF (AND(Serial_port.PROTOCOL, 16#0F) = 06)
THEN (* modalità Modbus Master OK *)
Function_Step := 50; (* passo successivo *)
ELSE
Test_inc := Test_inc + 1;
IF (Test_inc > 1000) THEN
Error := 45; (* errore *)
Function_Step := 130; (* passo successivo = fine *)
END_IF;
END_IF;
ELSE (* errore d’invio comando alla porta *)
Error := 45; (* errore *)
Function_Step := 130;
END_IF;
END_IF;

50: (*Scrittura di informazioni nella seconda CPU*)
Mngtwrite[2]:=50; (* timeout*)
%MW40:=5; (* valore da inviare *)
WRITE_VAR(ADDM(Adr_modbus),’%MW’,100,2,%MW40:2,Mngtwrite);
Function_Step := 55;

55: (* Risultato della funzione Test WRITE_VAR *)
IF (NOT Mngtwrite[0].0) THEN
IF (Mngtwrite[1] = 0) THEN
Function_Step := 60; (* success : passo successivo *)
ELSE
Error := 55; (* errore *)
Function_Step := 130; (* passo successivo = fine *)
END_IF;
END_IF;
```

```
60: (* Invia comando per commutare porta seriale dalla modalità Modbus a Modalità caratteri *)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* nessun comando attivo *)
 Serial_port.CONTROL := 16#00; (* reset parola di controllo *)
 (* imposta bit comando TO_CHAR_MODE *)
 SET(Serial_port.TO_CHAR_MODE);
 WRITE_CMD (Serial_port); (* invio comando *)
 Test_inc := 0; (* inizializza Riprova conteggio *)
 Function_Step := 65; (* passo successivo *)
 END_IF;

65: (* Risultato test del comando di commutazione *)
 READ_STS(Serial_port); (* lettura stato porta seriale *)
 IF (Serial_port.EXCH_STS = 0) THEN (* comando completato *)
 (* reset bit di comando TO_CHAR_MODE *)
 RESET(Serial_port.TO_CHAR_MODE);
 IF (Serial_port.EXCH_RPT = 0) THEN (* nessun errore *)
 IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)
 THEN (* modalità caratteri OK *)
 Function_Step := 70; (* passo successivo *)
 ELSE
 Test_inc := Test_inc + 1;
 IF (Test_inc > 1000) THEN
 Error := 65; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
 ELSE (* errore d’invio comando alla porta *)
 Error := 65; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
```

```
70: (* Disconnessione modem: passo 1*)
 ReqString := '+++'; (* sequenza escape *)
 MngtPrint[3] := 3; (* dimensione scambio in byte *)
 PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
 Function_Step := 75; (* passo successivo *)

75: (* Risultato funzione test PRINT_CHAR *)
 IF (NOT MngtPrint[0].0) THEN
 IF (MngtPrint[1] = 0) THEN
 (* Success : passo successivo *)
 Function_Step := 80;
 ELSE
 (* Fine con errore *)
 Error := 75;
 Function_Step := 130;
 END_IF;
 END_IF;
80:
 MngtInput[2] := 50; (* timeout *)
 INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput, AnsString2); (*Attesa
risposta modem*)
 Function_Step := 85; (* passo successivo*)
85: (* Risultato funzione test INPUT_CHAR *)
 IF (NOT MngtInput[0].0) THEN
 IF (MngtInput[1] = 0) THEN
 (* Success : passo successivo *)
 Function_Step := 90;
 ELSE
 (* Fine con errore *)
 Error := 85;
 Function_Step := 130;
 END_IF;
 END_IF;
```

```
90: (* Risposta test modem *)
 IF (AnsString2 = '$NOK') THEN
 Function_Step := 95; (* success : passo successivo *)
 ELSE
 Error := 90; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
95: (* Disconnessione modem: passo 2*)
 ReqString := 'ATH0$N'; (* chiusura messaggio *)
 MngtPrint[3] := 3; (* dimensione scambio in byte *)
 PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
 Function_Step := 100; (* passo successivo *)
100: (* Risultato funzione test PRINT_CHAR *)
 IF (NOT MngtPrint[0].0) THEN
 IF (MngtPrint[1] = 0) THEN
 (* Success : passo successivo *)
 Function_Step := 105;
 ELSE
 (* Fine con errore *)
 Error := 100;
 Function_Step := 130;
 END_IF;
 END_IF;
105:
 MngtInput[2] := 50; (* timeout *)
 INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput, AnsString3); (*Attesa
risposta modem*)
 Function_Step := 110; (* passo successivo*)
110: (* Risultato funzione test INPUT_CHAR *)
 IF (NOT MngtInput[0].0) THEN
 IF (MngtInput[1] = 0) THEN
 (* Success : passo successivo *)
 Function_Step := 115;
 ELSE
```

```
(* Fine con errore *)
Error := 110;
Function_Step := 130;
END_IF;
END_IF;
115: (* Risposta test modem *)
IF (AnsString3 = ’$NOK’) THEN
Function_Step := 120; (* success : passo successivo *)
ELSE
Error := 115; (* errore *)
Function_Step := 130; (* passo successivo = fine *)
END_IF;
120: (* Invio comando di commutazione porta seriale dalla Modalità caratteri a modalità Slave
Modbus *)
READ_STS(Serial_port); (* lettura stato porta seriale *)
IF (Serial_port.EXCH_STS = 0) THEN (* nessun comando attivo *)
Serial_port.CONTROL := 16#00; (* reset parola di controllo *)
(* imposta bit di comando TO_MODBUS_SLAVE *)
SET(Serial_port.TO_MODBUS_SLAVE);
WRITE_CMD (Serial_port); (* invio comando *)
Test_inc := 0; (* inizializza Riprova conteggio *)
Function_Step := 125; (* passo successivo *)
END_IF;

125: (* Risultato test del comando di commutazione *)
READ_STS(Serial_port); (* lettura stato porta seriale *)
IF (Serial_port.EXCH_STS = 0) THEN (* comando completato *)
(* reset bit di comando TO_MODBUS_SLAVE *)
RESET(Serial_port.TO_MODBUS_SLAVE);
IF (Serial_port.EXCH_RPT = 0) THEN (* nessun errore *)
IF (AND(Serial_port.PROTOCOL, 16#0F) = 07)
THEN (* modalità caratteri OK *)
Function_Step := 130; (* passo successivo *)
ELSE
```

```
 Test_inc := Test_inc + 1;
 IF (Test_inc > 1000) THEN
 Error := 125; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
 ELSE (* errore d’invio comando alla porta *)
 Error := 125; (* errore *)
 Function_Step := 130; (* passo successivo = fine *)
 END_IF;
 END_IF;
130: (* Fine *)
 IF (NOT Start) THEN (* flag attivatore *)
 Function_Step := 0; (* vai a stato di attesa *)
 END_IF;
END_CASE;
```

Capitolo 14
Avvio dell’applicazione

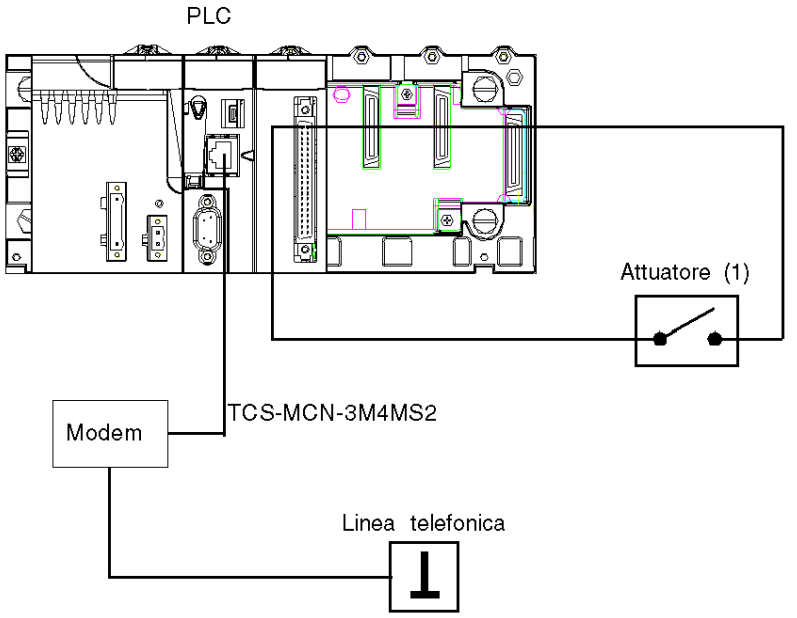
Esecuzione dell’applicazione in modalità standard

In breve

In questo esempio, il funzionamento in modalità standard richiede l’uso di due PLC, di un modulo di ingresso digitale, di un modulo BMX NOM 0200 e di 2 modem SR2MOD01.

Cablaggio primo PLC slave

Il primo PLC slave è collegato come segue:



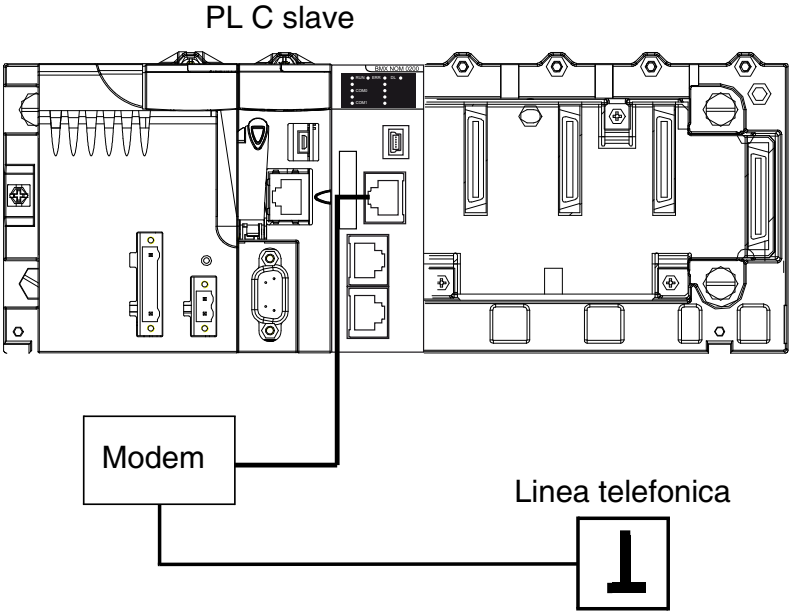
(1): l'attuatore è connesso al canale 0 del modulo digitale.

In questo esempio, il primo modem viene collegato alla porta del processore seriale del primo PLC slave.

Lo stato dell’attatuatore controlla lo stato della variabile `Start` nell’applicazione.

Cablaggio secondo PLC slave

Il secondo PLC slave è collegato come segue:



In questo esempio, il secondo modem è collegato al canale 0 del modulo BMX NOM 0200 del secondo PLC slave.

Per una migliore affidabilità nella comunicazione, il cavo TCS XCN 3M4F3S4 viene utilizzato per i segnali del modem DTR/DSR/DCD gestiti dall'applicazione.

Configurazione del secondo PLC slave

Prima di trasferire il progetto per la configurazione del secondo PLC slave, accertarsi che il secondo PLC slave non sia collegato al modem.

Nella seguente tabella è descritta la procedura per trasferire il progetto in modalità standard:

Passo	Azione
1	Nel menu `PLC`, fare clic su `Modalità standard`.
2	Nel menu `Creazione`, fare clic su `Ricrea tutto il progetto`. Il progetto viene generato ed è pronto per essere trasferito al PLC.
3	Nel menu `PLC`, fare clic su `Connetti`. Viene eseguita la connessione al PLC.
4	Nel menu `PLC`, fare clic su `Trasferimento progetto al PLC`. Viene visualizzata la finestra `Trasferimento progetto al PLC`. Fare clic su `Trasferisci`. L'applicazione viene trasferita al PLC.
5	Collegare il secondo PLC slave a un modem SR2MOD01.

Trasferimento dell'applicazione al primo PLC slave

Prima di trasferire l'applicazione, accertarsi che il primo PLC slave non sia collegato al modem.

Nella seguente tabella è descritta la procedura per trasferire l'applicazione in modalità standard:

Passo	Azione
1	Nel menu `PLC`, fare clic su `Modalità standard`.
2	Nel menu `Creazione`, fare clic su `Ricrea tutto il progetto`. Il progetto viene generato ed è pronto per essere trasferito al PLC. Quando si genera il progetto, viene visualizzata la finestra dei risultati. Se nel programma si verifica un errore, Unity Pro ne indica la posizione (facendo clic sulla sequenza evidenziata).
3	Nel menu `PLC`, fare clic su `Connetti`. Viene eseguita la connessione al PLC.
4	Nel menu `PLC`, fare clic su `Trasferimento progetto al PLC`. Viene visualizzata la finestra `Trasferimento progetto al PLC`. Fare clic su `Trasferisci`. L'applicazione viene trasferita al PLC.

Esecuzione della prima applicazione sul primo PLC slave

Nella seguente tabella è descritta la procedura per eseguire l'applicazione in modalità standard:

Passo	Azione
1	Nel menu `PLC`, fare clic su `Run`. Viene visualizzata la finestra `Run`. Fare clic su `OK`. L'applicazione è in esecuzione sul PLC.
2	Scollegare il PC sul quale viene eseguito il software Unity Pro dal primo PLC slave.
3	Collegare il primo PLC slave a un modem SR2MOD01.

Glossario

0-9

%I

Secondo lo standard CEI, %I indica un oggetto linguaggio di tipo ingresso digitale.

%IW

Secondo lo standard CEI, %I indica un oggetto linguaggio di tipo ingresso analogico.

%KW

Secondo lo standard CEI, %KW indica un oggetto linguaggio di tipo parola costante.

%M

Secondo lo standard CEI, %M indica un oggetto linguaggio di tipo bit memoria.

%MW

Secondo lo standard CEI, %MW indica un oggetto linguaggio di tipo parola memoria.

%Q

Secondo lo standard CEI, %Q indica un oggetto linguaggio di tipo uscita digitale.

%QW

Secondo lo standard CEI, %QW indica un oggetto linguaggio di tipo uscita analogica.

A

Altivar

Azionamento a velocità variabile AC.

ARRAY

Un ARRAY è una tabella di elementi dello stesso tipo. La sintassi è la seguente: ARRAY [<limiti>] OF <Tipo>Esempio:ARRAY [1..2] OF BOOL è una tabella unidimensionale con due elementi di tipo BOOL.ARRAY [1..10, 1..20] OF INT è una tabella bidimensionale con elementi 10x20 di tipo INT.

ASCII

ASCII è l'abbreviazione di American Standard Code for Information Interchange. È uno standard americano, diventato standard internazionale, che utilizza 7 bit per definire tutti i caratteri alfanumerici utilizzati nella lingua inglese, i caratteri di punteggiatura, alcuni caratteri grafici e vari altri comandi.

B

BOOL

BOOL è l'abbreviazione del tipo booleano. Si tratta dell'elemento dati di base in informatica. Una variabile di tipo BOOL possiede uno dei due valori seguenti: 0 (FALSE) o 1 (TRUE). Un bit estratto di parola è di tipo BOOL, ad esempio: %MW10.4.

Broadcast

Le comunicazioni broadcast inviano pacchetti da una stazione a ogni destinazione di rete. I messaggi broadcast si riferiscono a ogni dispositivo di rete o solo ad un dispositivo per il quale l'indirizzo non è noto.

BYTE

Quando sono raggruppati 8 bit, si parla di BYTE. Un BYTE può essere immesso in modalità binaria o in base 8. Il tipo BYTE è codificato in un formato a 8 bit che, in formato esadecimale, va da 16#00 a 16#FF.

C

Configurazione

La configurazione raccoglie i dati che caratterizzano la macchina (invariante) e che sono necessari per il funzionamento del modulo. Tutte queste informazioni sono memorizzate nella zona costante PLC %KW. L'applicazione PLC non può modificarle.

CPU

CPU è l'acronimo di Central Processing Unit (Unità centrale di programmazione): nome generico per i processori Schneider Electric.

CRC

CRC è l'acronimo di Cyclic Redundancy Checksum: indica se un carattere è stato "deformato" durante la trasmissione dei frame.

D

DFB

DFB è l'acronimo di Derived Function Block (blocco funzione derivato). I tipi DFB sono blocchi funzione che possono essere definiti dall'utente in linguaggio ST (Structured Text), IL (Instruction List), LD (Ladder Diagram) o FBD (Function Block Diagram). L'uso di questi tipi DFB in un'applicazione consente di:

- semplificare la concezione e l'immissione del programma;
- accrescere la leggibilità del programma;
- facilitare il debug;
- diminuire il volume del codice creato.

DINT

DINT è l'abbreviazione del formato Double INTeger (intero doppio) (codificato a 32 bit). I limiti inferiore e superiore sono i seguenti: da -(2 alla potenza di 31) a (2 alla potenza di 31) - 1.Esempio:-2147483648, 2147483647, 16#FFFFFFFF.

E

EBOOL

EBOOL è l'abbreviazione del tipo Extended BOOLean (booleano esteso). Una variabile di tipo EBOOL ha valore (0 (FALSE) o 1 (TRUE) e dispone di fronti di salita o di discesa e di funzioni di forzatura. Una variabile di tipo EBOOL occupa un byte di memoria. Il byte è composto come segue:

- un bit per il valore;
- un bit per la cronologia (ogni volta che l'oggetto di stato cambia, il valore viene copiato nel bit di cronologia);
- un bit per la forzatura (uguale a 0 se l'oggetto non è forzato, uguale a 1 se il bit è forzato).

Il valore predefinito di ogni bit è 0 (FALSE).

EF

EF è l'abbreviazione di Elementary Function (funzione elementare). Si tratta del blocco, utilizzato in un programma, che esercita una funzione software predefinita. Una funzione non dispone di informazioni sullo stato interno. Più chiamate della stessa funzione con gli stessi parametri di ingresso forniranno sempre gli stessi valori di uscita. Informazioni sulla forma grafica della chiamata di funzione si trovano nel ["blocco funzionale (istanza)]". A differenza della chiamata dei blocchi funzione, le chiamate di funzione comportano solo un'uscita che non è nominata e il cui nome è identico a quello della funzione. In FBD, ogni chiamata è indicata da un [numero] unico mediante il blocco grafico. Questo numero viene generato automaticamente e non è modificabile. L'utente posiziona e configura queste funzioni nel programma per poter eseguire l'applicazione. Può anche sviluppare altre funzioni con il kit di sviluppo SDKC.

F

FBD

FBD è l'acronimo di Function Block Diagram (Diagramma blocco funzione). FDB è un linguaggio di programmazione grafica che funziona come un logigramma. A complemento dei blocchi logici semplici (AND, OR, ecc.), ogni funzione o blocco funzione del programma è rappresentato in questa forma grafica. Per ogni blocco, gli ingressi si trovano a sinistra e le uscite a destra. È possibile collegare le uscite dei blocchi agli ingressi di altri blocchi per formare espressioni complesse.

Fipio

Bus di campo utilizzato per collegare i dispositivi tipo sensore o attuatore.

Frame

Un frame è un gruppo di bit che forma un blocco digitale di informazioni. I frame contengono informazioni o dati di controllo della rete. La dimensione e la composizione di un frame è determinata dalla tecnologia di rete utilizzata.

Full duplex

Un metodo di trasmissione dati capace di trasmettere o ricevere sullo stesso canale simultaneamente.

H

Half duplex

Un metodo di trasmissione dati capace di comunicazione in due direzioni, ma solo in una direzione alla volta.

Hub

Un dispositivo hub che collega una serie di moduli flessibili e centralizzati per creare una rete.

I

Indirizzo

Su una rete, l'indentificazione di una stazione. In un frame, il raggruppamento di bit che identifica la sorgente o destinazione del frame.

INT

INT è l'abbreviazione del formato single INTeger (intero semplice) (codificato a 16 bit). I limiti inferiore e superiore sono i seguenti: da -(2 alla potenza di 15) a (2 alla potenza di 15) - 1.Esempio:-32768, 32767, 2#1111110001001001, 16#9FA4.

IODDT

IODDT è l'acronimo di Input/Output Derived Data Type (Tipo dati derivato di I/O). Il termine IODDT indica un tipo di dati strutturato che rappresenta un modulo o un canale di un modulo PLC. Ogni modulo esperto dispone di IODDT propri.

L

LED

LED è l'acronimo di Light emitting diode (diode emittente luce). Un indicatore che si illumina quando l'elettricità passa attraverso di esso. Indica lo stato di funzionamento di un modulo di comunicazione.

LRC

LRC è l'acronimo di Longitudinal redundancy check: è stato progettato per risolvere la bassa probabilità di rilevamento errore del controllo parità.

M

memoria FLASH

La memoria FLASH è una memoria non volatile che può essere sovrascritta. Viene mantenuta in una particolare EEPROM che può essere cancellata e riprogrammata.

Modulo digitale

Modulo con ingressi/uscite digitali.

Momentum

Moduli I/O che usano diverse reti di comunicazione standard aperte.

P

PLC

PLC è l'acronimo di Programmable logic controller. Il PLC è il centro di elaborazione di un processo di produzione industriale. Automatizza un processo, al contrario di quanto avviene nei sistemi di controllo a relè. I PLC sono computer previsti per operare nelle condizioni critiche tipiche degli ambienti industriali.

Presa

L'associazione di una porta con un indirizzo IP che funziona come identificazione di mittente o ricevente.

Protocollo

Descrive i formati dei messaggi e imposta un insieme di regole utilizzate da due o più dispositivi per comunicare utilizzando tali formati.

R

Rete

Il termine "rete" può avere due significati.

- In LD (Ladder Diagram): una rete è una serie di elementi grafici interconnessi. La portata di una rete è locale rispetto all'unità organizzativa del programma (sezione) che contiene la rete.
- Con i moduli esperti di comunicazione: una rete è una serie di stazioni che comunicano tra di loro. Il termine rete è utilizzato anche per definire un gruppo di elementi grafici interconnessi. Questo gruppo costituisce parte di un programma che può essere composto da un gruppo di reti.

RS232

Standard di comunicazione seriale che definisce la tensione del seguente servizio:

- un segnale di +12 V indica uno 0 logico
- un segnale di -12 V indica un 1 logico.

Tuttavia, in caso di attenuazione del segnale, il rilevamento avviene fino al limite di -3 V e +3 V. Tra questi due limiti, il segnale può essere considerato non valido. I collegamenti RS232 sono assai sensibili alle interferenze. Lo standard specifica di non superare una distanza di 15 m o un massimo di 9600 baud (bit/s).

RS485

Standard di collegamento seriale che opera con un differenziale di 10 V/+5 V. Utilizza due fili per l'invio e la ricezione dei dati. Le uscite "a 3 stati" permettono di passare in modalità di ascolto una volta che la trasmissione è terminata.

RTU

RTU è l'acronimo di Remote Terminal Unit. In modalità RTU, i dati vengono inviati come due bit da 4, caratteri esadecimali, fornendo una velocità di trasmissione superiore rispetto alla modalità ASCII per la stessa velocità baud. Il Modbus RTU è un protocollo binario e più critico al ritardo temporale rispetto al protocollo ASCII.

S

SEPAM

Relè di protezione digitale per protezione, controllo e monitoraggio di sistemi di alimentazione.

Sezione

Modulo di programma appartenente a un task che può essere scritto nel linguaggio scelto dal programmatore (FBD, LD, ST, IL, o SFC). Un task può essere costituito da più sezioni, il cui ordine di esecuzione corrisponde all'ordine in cui sono create. Questo ordine è modificabile.

ST

ST è l'acronimo di Structured Text. Il linguaggio letterale strutturato è un linguaggio elaborato simile ai linguaggi informatici di programmazione. Consente di strutturare serie di istruzioni.

STRING

Una variabile STRINGA è una serie di caratteri ASCII. La lunghezza massima di una stringa è 65.534 caratteri.

T

TAP

TAP è l'acronimo di Transmission Access Point: l'unità di collegamento del bus.

Task

È un gruppo di sezioni o subroutine eseguite ciclicamente o periodicamente dal task MAST, o periodicamente per il task FAST. Un task possiede un livello di priorità ed è collegato agli ingressi e alle uscite del PLC. Questi I/O vengono aggiornati di conseguenza.

Task Master

È il task principale del programma. È obbligatorio ed è utilizzato per eseguire l'elaborazione sequenziale del PLC.

U

Unity Pro

Software di programmazione PLC Schneider Automation.

V

Variabile

Entità di memoria del tipo BOOL, WORD, DWORD e così via, il cui contenuto è modificabile dal programma in esecuzione.

W

WORD

Il tipo WORD è codificato in un formato a 16 bit ed è utilizzato per eseguire elaborazioni in stringhe di bit.

Questa tabella fornisce i limiti inferiore/superiore delle basi che è possibile utilizzare:

Base	Limite inferiore	Limite superiore
Esadecimale	16#0	16#FFFF
Ottale	8#0	8#177777
Binario	2#0	2#1111111111111111

Esempi di rappresentazione:

Dati	Rappresentazione in una delle basi
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

X

XBT

Terminale operatore grafico.

XPS

Modulo di sicurezza utilizzato per l’elaborazione dei segnali di sicurezza per monitorare sia i componenti che i cablaggi di un sistema di sicurezza, compresi i dispositivi per il monitoraggio generico e i modelli di applicazione specifica.

Indice analitico

A

Accessori di cablaggio, *60*
Avvio rapido, *249*

B

BMXNOM0200, *19*
BMXP341000, *19*
BMXP342000, *19*
BMXP342010, *19*
BMXP3420102, *19*
BMXP342020, *19*

C

Cablaggio, *60*
Configurazione modalità caratteri, *106*
Configurazione Modbus, *77*

D

debug Modbus, *166*
Debug Modbus, *98*
Debugging modalità caratteri, *123*

I

Impostazioni dei parametri, *209*
INPUT_CHAR, *117*, *188*

M

M340
    hardened, *32*
    versione rinforzata, *32*
Messa a terra, *33*
modalità carattere, *101*
Modbus bus, *69*
Modifica dei protocolli, *246*
modifica protocolli, *244*

P

periferiche di connessione, *39*
PRINT_CHAR, *117*, *188*
Programmazione del bus Modbus, *88*
Programmazione modalità caratteri, *117*

S

struttura dati canale per la comunicazione modbus
    T_COM_CHAR_BMX, *232*
Struttura dati del canale per comuncazioni in modalità caratteri
    T_COM_CHAR_BMX, *233*
Struttura dati del canale per tutti i moduli
    T_GEN_MOD, *238*
Struttura dei dati del canale per la comunicazione Modbus
    T_COM_MB_BMX, *224*, *225*
Struttura dei dati del canale per protocolli di comunicazione
    T_COM_STS_GEN, *219*, *220*
Struttura dei dati del canale per tutti i moduli
    T_GEN_MOD, *238*

T

T_COM_CHAR_BMX, *232*, *233*
T_COM_MB_BMX, *224*, *225*
T_COM_STS_GEN, *219*, *220*
T_GEN_MOD, *238*, *238*
T_M_COM_NOM, *240*